# Biased versus unbiased numerical methods for stochastic simulations

Check for updates

Javier Aguilar [1,2] ✉, José J. Ramasco [1] & Raúl Toral [1]

Approximate numerical methods are one of the most used strategies to extract information from many-interacting-agents systems. In particular, numerical approximations are of extended use to deal with epidemic, ecological and biological models, since unbiased methods like the Gillespie algorithm can become unpractical due to high CPU time usage required. However, the use of approximations has been debated and there is no clear consensus about whether unbiased methods or biased approach is the best option. In this work, we derive scaling relations for the errors in approximations based on binomial extractions. This finding allows us to build rules to compute the optimal values of both the discretization time and number of realizations needed to compute averages with the biased method with a target precision and minimum CPU-time usage. Furthermore, we also present another rule to discern whether the unbiased method or biased approach is more efficient. Ultimately, we will show that the choice of the method should depend on the desired precision for the estimation of averages.

Epidemic modeling has traditionally relied on stochastic methods to go beyond mean-field deterministic solutions[1–5]. The contagion process itself is naturally adapted to a stochastic treatment since the basic units, individuals, can not be described successfully using deterministic laws. For example, two given individuals may or may not develop a contact even though they are potentially able to do so given their geographical location. Even further, should the contact be established and should one of the individuals be infectious, the infection of the second individual is not a certainty, but rather an event that occurs with some probability. Computational epidemiologists have implemented these stochastic contagions in all the modeling efforts and at different scales, from agent-based[6–12] to population-based[13–17]. In the case of agent-based models stochastic contagion events can be traced one by one, even though for practical purposes in computation sometimes they may be aggregated. In the population-level models, different contagion processes are aggregated together following some generic feature (number of neighboring individuals, geographical location, etc.). These models have the virtue of drastically reducing the number of variables needed to describe the whole population and, at the computational level, the positive side effect of enormously decreasing the model running time. A widespread practice nowadays[18–25] is to approximate the statistical description of these contagion aggregations by binomial or multinomial distributions.

The same methodological and conceptual issues appear well beyond the reign of epidemic modeling. Indeed, stochastic processes are one of the main pillars of complexity science[26–28]. The list of fruitful applications is endless, but just to name a few paradigmatic examples: population dynamics in ecology[29,30], gene expression[31], metabolism in cells[32], finances and market crashes[33,34], telecommunications[35], chemical reactions[36], quantum physics[37] and active matter[38]. As models become more intricate, there arises the technical challenge of producing stochastic trajectories in feasible computation times, since unbiased methods that generate statistically correct realizations of stochastic trajectories may become unpractical due to lengthy computations. Approximate methods aim at solving this issue by significantly reducing the CPU time usage. The use of approximated methods is extended (see e.g.[16,17,21]), and some authors assert that they might be the only way to treat effectively large systems of heterogeneous agents[39]. However, other works claim that the systematic errors induced by the approximations might not trade-off the reduction in computation time[40,41]. The primary objective of this work is to shed light on this debate and assess in which circumstances approximate methods based on binomial extractions, which we call *binomial methods*, can be advantageous with respect to the unbiased algorithms.

To solve this question, we derive in this paper a scaling relation for the errors of the binomial methods. This main result allows us to obtain optimal values for the discretization time and number of realizations to compute averages with a desired precision and minimum CPU time consumption. Furthermore, we derive a rule to discern if the binomial method is going to

[1]Instituto de Física Interdisciplinar y Sistemas Complejos IFISC (CSIC-UIB), Campus UIB, 07122 Palma de Mallorca, Spain. [2]Investigador ForInDoc del Govern de les Illes Balears en el departamento de Electromagnetismo y Física de la Materia e Instituto Carlos I de Física Teórica y Computacional, Universidad de Granada, Granada E-18071, Spain. ✉e-mail: javieraguilar@ifisc.uib-csic.es

be faster than the unbiased counterparts. Lastly, we perform a numerical study to compare the performance of both the unbiased and binomial methods and check the applicability of our proposed rules. Ultimately, we will show that the efficiency of the binomial method is superior to the unbiased approaches only when the target precision is below a certain threshold value.

## Methods

### Transition rates

Throughout this work we will focus on pure jumping processes, this is, stochastic models in which agents switch states within a discrete set of possible states. Spontaneous creation or annihilation of agents will not be considered, therefore, its total number, $N$, is conserved. We furthermore assume Markovian dynamics, so given that the system is in a particular state at time $t$, the microscopic rules that dictate the switching between states just depend on the current state $\boldsymbol{s}(t) = \{s_1(t), \ldots, s_N(t)\}$. These microscopic rules are given in terms of the transition rates, defined as the conditional probabilities per unit of time to observe a transition,

$$w^t(\boldsymbol{s} \to \boldsymbol{s}') := \lim_{dt \to 0} \frac{P(\boldsymbol{s}'; t + dt | \boldsymbol{s}; t)}{dt}. \tag{1}$$

A particular set of transitions in which we are specially interested define the one-step processes, meaning that the only transitions allowed are those involving the change of a single agent's state, with rates

$$w_i^t(s_i \to s_i') := w^t(\{s_1, \ldots, s_i, \ldots, s_N\} \to \{s_1, \ldots, s_i', \ldots, s_N\}), \tag{2}$$

for $i = 1, \ldots, N$. Our last premise is to consider only transition rates $w_i(s_i \to s_i')$ that do not depend explicitly on time $t$. Note that the rates could, in principle, be different for every agent and depend in an arbitrary way on the state of the system. The act of modeling is actually to postulate the functional form of these transition rates. This step is conceptually equivalent to the choice of a Hamiltonian in equilibrium statistical mechanics.

Jumping processes of two-state agents, such that the possible states of the $i^{th}$ agent can be $s_i = 0$ or $s_i = 1$, are widely used in many different applications, such as protein activation[42], spins 1/2[43], epidemic spreading[4,44], voting dynamics[45], chemical reactions[46,47], drug-dependence in pharmacology[48], etc. For binary-state systems, quite commonly, the rate of the process $s_i = 0 \to s_i = 1$ is different from the reverse process $s_i = 1 \to s_i = 0$ and we define the rate of agent $i$ as

$$w_i(s_i) := \begin{cases} w_i(0 \to 1) & \text{if } s_i = 0, \\ w_i(1 \to 0) & \text{if } s_i = 1. \end{cases} \tag{3}$$

As a detailed observation is usually unfeasible, we might be interested in a macroscopic level of description focusing, for example, on the occupation number $n(t)$, defined as the total number of agents in state 1,

$$n(t) := \sum_{i=1}^{N} s_i(t), \tag{4}$$

being $N - n(t)$ the equivalent occupation of state 0. In homogeneous systems, those in which $w_i(s_i) = w(s_i), \ \forall i$, transition rates at this coarser level can be computed from those at the agent level as

$$W(n \to n + 1) = (N - n)w(0),$$
$$W(n \to n - 1) = nw(1). \tag{5}$$

Some applications might require an intermediate level of description between the fully heterogeneous [Eq. (2)] and the fully homogeneous [Eq. (5)]. In order to deal with a coarse-grained heterogeneity, we define $\mathcal{C}$ different classes of agents. Agents can be labeled in order to identify their class, so that $l_i = \ell$ means that the $i^{th}$ agent belongs to the class labeled $\ell$ with $\ell \in [1, \mathcal{C}]$ and we require that all

agents in the same class share the same transition rates $w_i(s_i) = w_\ell(s_i), \ \forall l_i = \ell$. This classification allows us to define the occupation numbers $N_\ell$ and $n_\ell$ as the total number of agents of the $\ell^{th}$ class and the number of those in state 1 respectively. Moreover, we can write the class-level rates:

$$W_\ell(n_\ell \to n_\ell + 1) = (N_\ell - n_\ell)w_\ell(0),$$
$$W_\ell(n_\ell \to n_\ell - 1) = n_\ell w_\ell(1). \tag{6}$$

In general, stochastic models are very difficult and can not be solved analytically. Hence, one needs to resort to numerical simulations than can provide suitable estimations to the quantities of interest. There are two main types of simulation strategies: unbiased continuous-time and discrete-time algorithms. Each one comes with its own advantages and disadvantages that we summarize in the next sections.

### Unbiased continuous-time algorithms

We proceed to summarize the main ideas behind the unbiased continuous-time algorithms, and refer the reader to[39,44,49–54] for a detailed description. Say that we know the state of the system $\boldsymbol{s}(t)$ at a given time $t$. Such a state will remain unchanged until a random time $t' > t$, when the system experiences a transition or "jump" to a new state, also random, $\boldsymbol{s}'(t')$:

$$\boldsymbol{s}(t) \overset{t'-t}{\to} \boldsymbol{s}'(t'). \tag{7}$$

Therefore, the characterization of a change in the system necessarily requires us to sample both the transition time $\Delta t = t' - t$ and the new state $\boldsymbol{s}'(t')$.

For binary one-step processes, new states are generated by changes in single agents $s_i \to 1 - s_i$. The probability that agent $i$ changes its state in a time interval $t' \in [t, t + dt]$ is $w_i(s_i)dt$ by definition of transition rate. Therefore, the probability that the agent will not experience such a transition in an infinitesimal time interval is $1 - w_i(s_i)dt$. Concatenating such infinitesimal probabilities, we can compute the probability $Q_i(s_i, \Delta t)$ that a given agent does not change its state during an arbitrary time lapse $\Delta t$ as well as the complementary probability $P_i(s_i, \Delta t)$ that it does change state as

$$Q_i(s_i, \Delta t) = \lim_{dt \to 0}(1 - w_i(s_i)dt)^{\Delta t/dt} = e^{-w_i(s_i)\Delta t},$$
$$P_i(s_i, \Delta t) = 1 - e^{-w_i(s_i)\Delta t}. \tag{8}$$

Eq. (8) conforms the basic reasoning from which most of the continuous-time algorithms to simulate stochastic trajectories are built. It allows us to extend our basic postulate from Eq. (1), which only builds probabilities for infinitesimal times ($dt$), to probabilities of events of arbitrary duration ($\Delta t$). We note that Eq. (8) is actually a conditional probability: it is only valid provided that there are no other updates of the system in the interval $\Delta t$. From it we can also compute the probability density function that the $i^{th}$ agent remains at $s_i$ for a non-infinitesimal time $\Delta t$ and then experiences a transition to $s_i' = 1 - s_i$ in the time interval $[t + \Delta t, t + \Delta t + dt]$:

$$f_i(s_i; \Delta t) = e^{-w_i(s_i)\Delta t}w_i(s_i). \tag{9}$$

The above quantity is also called first passage distribution for the $i^{th}$ agent. Therefore, given that the system is in state $\boldsymbol{s}$ at time $t$, one can use the elements defined above to compute the probability that the next change of the system is due to switching in the agent $i$ at time $t' \in [t + \Delta t, t + \Delta t + dt]$:

$$\begin{aligned} &\text{P}(i^{th} \text{ agent switches state in } [t + \Delta t, t + \Delta t + dt]) \\ &\times \text{P}(\text{Other agents change state only after } t + \Delta t + dt) \\ &= f_i(s_i; \Delta t)dt \times \prod_{j \neq i}^{N} Q_j(s_j, \Delta t) = e^{-W(\boldsymbol{s})\Delta t}w_i(s_i)dt, \end{aligned} \tag{10}$$

where we have defined the total exit rate,

$$W(\boldsymbol{s}) := \sum_{i=1}^{N} w_i(s_i). \tag{11}$$

Two methods, namely the first-reaction method and the Gillespie algorithm, can be distinguished based on the scheme used to sample the random jumping time $t'$ and switching agent $i$ from the distribution specified in Eq. (10). The first-reaction method involves sampling one tentative random time per transition using $f_i(s_i; \Delta t)$ and then choosing the minimum among them as the transition time and reaction that actually occurs. In contrast, the Gillespie algorithm directly samples the transition time using the total rate $W(\boldsymbol{s})$ and then determines which transition is being activated. Depending on the algorithm used to randomly select the next reaction, the computational complexity of the unbiased methods can vary from linear to constant in the number of reactions (see e.g.[54]). Through the rest of the paper, we will use Gillespie algorithms with binary search in representation of unbiased methods.

## Discrete-time approximations
In this section, we consider algorithms which at simulation step $j$ update time by a constant amount, $t_{j+1} = t_j + \Delta t$. Note that the discretization step $\Delta t$ is no longer stochastic, and it has to be considered as a new parameter that we are in principle free to choose. Larger values of $\Delta t$ result in faster simulations since fewer steps are needed in order to access enquired times. Nevertheless, the discrete-time algorithms introduce systematic errors that grow with $\Delta t$.

**Discrete-synchronous**. It is possible to use synchronous versions of the process where all agents can potentially update their state at the same time $t_j$ using the probabilities $P_i(s_i, \Delta t)$ defined in Eq. (8) (see e.g.[52,55,56]).

**Algorithm 1**. **Discrete time synchronous agent level**
1: Increment time: $t_{j+1} = t_j + \Delta t$
2: Compute all probabilities $P_i(s_i, \Delta t)$, $i = 1, \ldots, N$, using Eq. (8).
3: For all agents, generate a uniform random number $\hat{u}_i \in [0, 1]$. If $\hat{u}_i < P_i(s_i, \Delta t)$ change the state $s_i \to 1 - s_i$.
4: go to 1.

We note that the use of synchronous updates changes the nature of the process since simultaneous updates were not allowed in the original continuous-time algorithms. Given that the probabilities $P_i(s_i, \Delta t)$ tend to zero as $\Delta t \to 0$, one expects to recover the results of the continuous-time asynchronous approach in the limit $\Delta t \to 0$. Nevertheless, users of this method should bear in mind that this approximation could induce discrepancies with the continuous-time process that go beyond statistical errors[57].

**Binomial method: two simple examples**. When building the class version of the synchronous agent level (Algorithm 1), one can merge together events with the same transition probability and sample the updates using binomial distributions. This is the basic idea behind the binomial method, which is of extended use in the current literature (e.g.[17,23,25,58,59]). Since references presenting this method are scarce, we devote a longer section to its explanation.

Let us start with a simple example. Say that we are interested in simulating the decay of $N$ radioactive nuclei. We denote by $s_i = 1$ that nucleus $i$ is non-disintegrated and by $s_i = 0$ the disintegrated state. All nuclei have the same time-independent decay rate $\mu$:

$$w_i(1 \to 0) = \mu, \quad w_i(0 \to 1) = 0. \tag{12}$$

This is, all nuclei can decay with the same probability $\mu dt$ in every time-bin of infinitesimal duration $dt$, but the reverse reaction is not allowed. This simple stochastic process leads to an exponential decay of the average number $n_t$ of active nuclei at time $t$ as $\langle n_t \rangle = N e^{-\mu t}$.

Using the rates (12), we can compute the probability that one nucleus disintegrates in a non-infinitesimal time $\Delta t$ [Eq. (8)],

$$p := P_i(1, \Delta t) = 1 - e^{-\mu \Delta t}, \quad \forall i. \tag{13}$$

Therefore every particle follows a Bernoulli process in the time interval $\Delta t$. That is, each particle decays with a probability $p$ and remains in the same state with a probability $1 - p$. As individual decays are independent of each other, the total number of decays in a temporal bin of duration $\Delta t$ follows a binomial distribution $\mathbf{B}(N, p)$,

$$P[n \text{ decays in } \Delta t] = \binom{N}{n} p^n (1 - p)^{N-n}. \tag{14}$$

The average of the binomial distribution is $\langle n \rangle = Np$ and its variance $\sigma^2[n] = Np(1 - p)$. This result invites us to draw stochastic trajectories with a recursive relation:

$$n_{t+\Delta t} = n_t - \Delta n_t, \tag{15}$$

where we denote by $\Delta n_t \sim \mathbf{B}(n_t, p)$ a random value drawn from the binomial distribution, with average value $\langle \Delta n_t \rangle = n_t p$, and we start from $n_0 = N$. In this simple example, it turns out that Eq. (15) does generate unbiased realizations of the stochastic process. From this equation we obtain

$$\langle n_{t+\Delta t} \rangle_B = \langle n_t \rangle_B - \langle \Delta n_t \rangle_B = \langle n_t \rangle_B (1 - p). \tag{16}$$

The symbol $\langle \cdot \rangle_B$ denotes averages over the binomial method. The solution of this recursion relation with initial condition $n_0 = N$ is

$$\langle n_t \rangle_B = N (1 - p)^{\frac{t}{\Delta t}} = N e^{-\mu t}, \tag{17}$$

which coincides with the exact result independently of the value of $\Delta t$. Therefore, the choice of $\Delta t$ is just related to the desired time resolution of the trajectories. If $\Delta t \ll (N\mu)^{-1}$, many of the outcomes $\Delta n_t$ used in Eq. (15) will equal zero as the resolution would be much smaller than the mean time between disintegration events. Contrary, if $\Delta t \gg (N\mu)^{-1}$, much of the information about the transitions will be lost and we would generate a trajectory with abrupt transitions. Still, both simulations would faithfully inform about the state of the system at the enquired times [see Fig. 1a, b].

Let us now apply this method to another process where it will no longer be exact. Nevertheless, the basic idea of the algorithm is the same: compute non-infinitesimal increments of stochastic trajectories using binomial distributions. We consider a system with $N$ agents which can jump between states with homogeneous constant rates:

$$w_i(1 \to 0) = \mu, \quad w_i(0 \to 1) = \kappa, \tag{18}$$

Which, at the macroscopic level read

$$W(n \to n - 1) = n\mu, \quad W(n \to n + 1) = (N - n)\kappa, \tag{19}$$

from which we can see that this process is a birth-death process. Reasoning as before, the probabilities that a particle changes state in a non-infinitesimal time $\Delta t$ are:

$$\begin{aligned} P(0, \Delta t) &= 1 - e^{-\kappa \Delta t}, \\ P(1, \Delta t) &= 1 - e^{-\mu \Delta t}. \end{aligned} \tag{20}$$

Where we can avoid the use of subscripts since all agents share the transition rates. At this point, we might feel also invited to write an equation for the evolution of agents in state 1 in terms of the stochastic number of transitions:

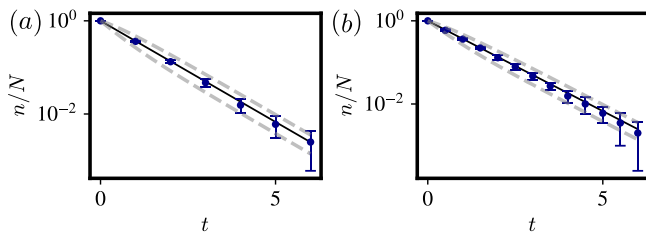$$n_{t+\Delta t} = n_t + \Delta n_{t,0} - \Delta n_{t,1}. \tag{21}$$

**Fig. 1 | Exact binomial method.** Simulations of the radioactive decay process with rates given by Eq. (12), using the binomial method [Eq. (15)]. In **a** the time discretization is $\Delta t = 1$, whereas in **b** is $\Delta t = 0.5$. In both panels $N = 100$ and $\mu = 1$. Dots and error bars indicate the average and standard error respectively, both computed from 20 simulations. With continuous line, we show the analytical average (black) plus and minus the analytical standard error (gray dashed lines): $\langle n(t)\rangle \pm \sigma[n(t)]/\sqrt{20}$. Independently of the discretization time, the results from simulations agree with the analytical value within errors.
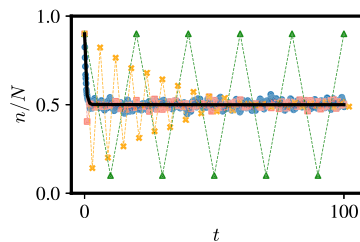


**Fig. 2 | Biased binomial method.** Four realizations of the birth and death process with constant rates defined by Eq. (18) simulated with the use of the binomial method [Eq. (21)]. In this case, we also use different time discretizations $\Delta t$, and fixed $N = 1000$, $\mu = 1$, and $\kappa = 1$. Note the numerical instability that shows up as wild oscillations in the numerical trajectories for large time steps $\Delta t = 10$ (triangles), and $\Delta t = 3$ (crosses). Otherwise, there is a good agreement between simulations and the expected average value (continuous black line) for both $\Delta t = 0.1, 1$ (circles and squares respectively).

Where $\Delta n_{t,0}$ and $\Delta n_{t,1}$ are binomial random variables distributed according to $\mathbf{B}(N - n_t, P(0, \Delta t))$ and $\mathbf{B}(n_t, P(1, \Delta t))$, respectively. However, trajectories generated with Eq. (21) turn out to be only an approximation to the original process. The reason is that the probability that a given number of transitions $0 \to 1$ happen in a time window is modified as soon as a transition $1 \to 0$ occurs (and vice-versa). If we now take averages in Eq. (21), use the known averages of the binomial distribution and solve the resulting linear iteration relation for $\langle n_t\rangle_B$, we obtain

$$\langle n_t\rangle_B = \left(n_0 - \frac{b}{a}\right)(1 - a)^{t/\Delta t} + \frac{b}{a} \tag{22}$$

with $a = 2 - e^{-\mu\Delta t} - e^{-\kappa\Delta t}$ and $b = N(1 - e^{-\kappa\Delta t})$. It is true that in the limit $\Delta t \to 0$, this solution recovers the exact solution for the evolution equation of the average number of agents in state 1 for the continuous-time process, namely

$$\frac{d\langle n_t\rangle}{dt} = -\mu\langle n_t\rangle + \kappa(N - \langle n_t\rangle),$$
$$\langle n_t\rangle = \left(n_0 - N\frac{\kappa}{\kappa + \mu}\right)e^{-(\kappa+\mu)t} + N\frac{\kappa}{\kappa + \mu}, \tag{23}$$

but the accuracy of the discrete approximation depends crucially on the value of $\Delta t$. If, for instance, we take $\Delta t \gg \max(\kappa^{-1}, \mu^{-1})$, then we can approximate $a \approx 2$, $b \approx N$, such that Eq. (22) yields

$$\langle n_t\rangle_B = \begin{cases} N - n_0, & \text{if } t/\Delta t \text{ odd,} \\ n_0, & \text{if } t/\Delta t \text{ even,} \end{cases} \tag{24}$$

a numerical instability that shows up as a wild oscillation, see Fig. 2.

Therefore, the fact that agents are independent and rates are constant is not sufficient condition to guarantee that the binomial method generates unbiased trajectories for arbitrary values of the discretization step $\Delta t$. Nevertheless, we note that the only condition needed to ensure that Eq. (21) is a good approximation to the exact dynamics, Eq. (23), is that $\Delta t \ll \min(\kappa^{-1}, \mu^{-1})$. Given than the system size $N$ does not appear in this condition, we expect the binomial method to be very efficient to simulate this kind of process if we take a sufficiently small value for $\Delta t$, independently of the number of agents, see Fig. 2, where both $\Delta t = 0.1, 1$ produce a good agreement for $\mu = \kappa = 1$. By comparing the average value of the binomial method, Eq. (22) with the exact value, Eq. (23), we note that the error of the binomial approximation can be expanded in a Taylor series

$$\langle n_t\rangle_B - \langle n_t\rangle = \lambda\Delta t + \mathcal{O}(\Delta t^2). \tag{25}$$

where the coefficient of the linear term $\lambda$ depends on $t$ and $N$, as well as on other parameters of the model. We will check throughout this work that a similar expansion of the errors in the binomial method holds for the case of more complex models.

**Binomial method: general algorithm.** If we go back to the general two-state process in which the functional form of the rates can have an arbitrary dependence on the state of the system, we can approximate the probability that the state of agent $i$ changes in a time interval $\Delta t$ by $P_i(s_i, \Delta t)$ [Eq. (8)]. If all these probabilities are different, we cannot group them in order to conform binomial samples. If, on the other hand, we can identify large enough classes $\ell = 1, 2, \ldots, C$ such that all $N_\ell$ agents in the same class $\ell$ have the same rates $w_\ell(s)$, we can approximate the variation of the occupation number $n_\ell$ of each class during the time $\Delta t$ as the difference $\Delta n_{\ell,0} - \Delta n_{\ell,1}$ where $\Delta n_{\ell,0}$ and $\Delta n_{\ell,1}$ follow, respectively, binomial distributions $B(N_\ell - n_\ell, P_\ell(0, \Delta t))$ and $B(n_\ell, P_\ell(1, \Delta t))$, with $P_\ell(s_i, \Delta t)$ given by Eq. (8) using any agent $i$ belonging to class $\ell$. All class occupation numbers are updated at the same time step $j$, yielding the synchronous binomial algorithm, which reads:

**Algorithm 2**. **Binomial synchronous class level**
  1: Update time as $t_{j+1} = t_j + \Delta t$.
  2: For every class $\ell \in [1, \ldots, C]$: Update the values of $P\ell(1, \Delta t)$, $P\ell(0, \Delta t)$, using Eq. (8).
  3: For every class $\ell \in [1, \ldots, C]$: Update the number of agents as $n_\ell \to n_\ell + \Delta n_{\ell,0} - \Delta n_{\ell,1}$, where $\Delta n_{\ell,0}$ and $\Delta n_{\ell,1}$ are values of binomial random variables distributed according to $\mathbf{B}(N_\ell - n_\ell, P_\ell(0, \Delta t))$ and $\mathbf{B}(n_\ell, P_\ell(1, \Delta t))$, respectively.
  4: go to 1.

A similar reasoning can be built departing from the knowledge that the number $n$ of occurrences of continuous-time independent processes with constant rates follows a Poisson distribution[36], namely $e^{-\Lambda}\Lambda^n/n!$, being the parameter $\Lambda$ of the Poisson distribution equal to the product of the rate times the time interval considered. Therefore, the number of firings of each class in the time interval $\Delta t$, $\Delta n_{\ell,1}$ and $\Delta n_{\ell,0}$, can be approximated by Poisson random variables with parameters $W_\ell(n_\ell \to n_\ell - 1)\Delta t$ and $W_\ell(n_\ell \to n_\ell + 1)\Delta t$, respectively. This conception gives rise to the $\tau$-leaping algorithm[39,44,56,60–63] used in the context of chemical modeling. Given that Poisson random variables are unbounded from above, the $\tau$-leaping algorithm may yield negative values for the occupation numbers $n_\ell$ (see e.g. refs. [39,56]). Consequently, our focus will be on the binomial method, which does not exhibit this drawback.

## Results and discussion
### The $\frac{27}{4}$ rule
The major drawback of the binomial method to simulate trajectories is the necessity of finding a proper discretization time $\Delta t$ that avoids both slow and inaccurate implementations. In this section, we propose a semi-empirical predictor for the values of the optimal choice of $\Delta t$ that propitiates the

smallest computation time for a fixed desired accuracy. Moreover, we will present a rule to discern whether an unbiased continuous-time algorithm or the discrete-time binomial method is more suitable for the required task.

Consider that we are interested in computing the average value $\langle Z \rangle$ of a random variable $Z$ that depends on the stochastic trajectory in a time interval $[0, T]$. For example, $Z$ could be the number of nuclei for the process defined in Eq. (12) at a particular time $t \in [0, T]$. We remark that $\langle Z \rangle$ could also stand for the central second moment of some random variable, thus accounting for fluctuations around some mean value. Also, the average $\langle Z \rangle$ could represent probabilities if $Z$ is chosen to be an indicator function (see e.g.[64]).

The standard approach to compute $\langle Z \rangle$ numerically generates $M$ independent realizations of the stochastic trajectories and measures the random variable $Z^{(i)}$ in each trajectory $i = 1, \ldots, M$. The average value $\langle Z \rangle$ is then approximated by the sample mean

$$Z_M := \frac{1}{M} \sum_{i=1}^{M} Z^{(i)}. \tag{26}$$

Note that $Z_M$ itself should be considered a random variable as its value changes from a set of $M$ realizations to another.

For an unbiased method, such as Gillespie, the only error $\varepsilon$ in the estimation of $\langle Z \rangle$ by $Z_M$ is of statistical nature and can be computed from the standard deviation of $Z_M$, namely

$$\varepsilon = \frac{\sigma}{\sqrt{M}}, \text{ with } \sigma := \sqrt{\langle Z^2 \rangle - \langle Z \rangle^2}. \tag{27}$$

The quantification of the importance of the error, for sufficiently large $M$, follows from the central limit theorem[50,64] using the confidence intervals of a normal distribution:

$$P\left[\langle Z \rangle - \varepsilon \le Z_M \le \langle Z \rangle + \varepsilon\right] = 0.6827\ldots \tag{28}$$

It is in this sense, that one says that the standard error $\varepsilon$ is the precision of the estimation and writes accordingly

$$\langle Z \rangle = Z_M \pm \varepsilon. \tag{29}$$

Note that, according to Eq. (27), for an unbiased method the error in the estimation of the sample mean $Z_M$ tends to zero in the limit $M \to \infty$.

For a biased method, such as the binomial, that uses a finite discretization time $\Delta t$ and generates $M_B$ independent trajectories, the precision is altered by a factor that does not tend to zero in the limit $M_B \to \infty$. Based on the result found in the simple birth and death example of the previous section, let us assume for now that this factor scales linearly with the discretization time $\Delta t$ and can be written as $\lambda \Delta t$ where $\lambda$ is a constant depending on the model. We will corroborate this linear assumption for the binomial method both with calculations and numerical simulations in the next section, and we refer to Supplementary Note 1 for a more general discussion in the case of a method with a possible non-linear dependence. Then we can write the estimator using the binomial method as

$$\langle Z \rangle = Z_{M_B} + \lambda \Delta t \pm \varepsilon_B, \tag{30}$$

where $\varepsilon_B = \frac{\sigma}{\sqrt{M_B}}$ and $Z_{M_B}$ is the sample average, Eq. (26), using $M_B$ realizations. The maximum absolute error of the biased method is then $|\lambda| \Delta t + \varepsilon_B$. Due to the presence of a bias term in the error, the only way that the precision of the binomial method can equal the one of an unbiased approach is by increasing the number of realizations $M_B$ compared to the number of realizations $M$ of the unbiased method. Matching the values of the errors of the unbiased and the biased methods, we arrive at the condition

that the required number of steps of the biased method is

$$M_B = M \left( \frac{|\lambda| \Delta t}{\varepsilon} - 1 \right)^{-2}, \tag{31}$$

and the additional requirement $\Delta t < \frac{\varepsilon}{|\lambda|}$ (otherwise the bias is so large that it can not be compensated by the increase in the number of realizations $M_B$).

What a practitioner needs is to compare the CPU times that the biased and unbiased methods require to achieve the same accuracy $\varepsilon$. For the biased method with a fixed time step $\Delta t$, the CPU time $t_B^{(CPU)}$ needed to generate one stochastic trajectory is proportional to the number of steps, $\frac{T}{\Delta t}$, needed to reach the final time $T$ and can be written as $C_B \frac{T}{\Delta t}$, where $C_B$ is the CPU time needed to execute one iteration of the binomial method. Hence the total time required to generate $M_B$ trajectories is

$$t_B^{(CPU)} = C_B M_B \frac{T}{\Delta t}. \tag{32}$$

(Note that in massive parallel architectures, it might be possible to obtain a sub-linear dependence of the total time with the number of realizations $M$. This possibility is discussed in Supplementary Note 1.)

The discretization time associated with a minimum value of the CPU time consumption and subject to the constraint of fixed precision is obtained by inserting Eq. (31) in Eq. (32) and minimizing for $\Delta t$ (see Supplementary Note 1). The optimal time reads:

$$\Delta t^{opt} = \frac{1}{3} \frac{\varepsilon}{|\lambda|}. \tag{33}$$

Inserting the equation for the optimal $\Delta t$ in Eq. (31), one obtains

$$M_B^{opt} = \frac{9}{4} M = \frac{9}{4} \left( \frac{\sigma}{\varepsilon} \right)^2, \tag{34}$$

which does not depend of $\lambda$ or other parameters. Eqs. (33) and (34) have major practical use, since they tell us how to choose $\Delta t^{opt}$ and $M_B^{opt}$ to use the binomial method to reach the desired precision $\varepsilon$ and with minimum CPU time usage.

Still, one important question remains. Provided that we use the optimal pair $(M_B^{opt}, \Delta t^{opt})$, is the binomial method faster than an unbiased approach? In order to answer this question we first obtain the expected CPU time of the binomial method with the optimal choice inserting Eqs. (33) and (34) in Eq. (32):

$$t_B^{(CPU,opt)} = \frac{27}{4} C_B \frac{|\lambda|}{\varepsilon} M T. \tag{35}$$

On the other hand, the CPU time needed to generate one trajectory using the unbiased method is proportional to the maximum time $T$, and the total CPU time to generate $M$ trajectories is $t_U^{(CPU)} = C_U M T$, where $C_U$ is a constant depending on the unbiased method used. The expected ratio between the optimal CPU time consumption with the binomial method an the unbiased approach is

$$\alpha = \frac{t_B^{(CPU,opt)}}{t_U^{(CPU)}} = \frac{27}{4} \frac{C_B}{C_U} \frac{|\lambda|}{\varepsilon}. \tag{36}$$

Equaion (36) defines what we called the $\frac{27}{4}$ rule, and its usefulness lies in the ability to indicate in which situations the binomial method is more efficient than the unbiased procedure (when $\alpha < 1$). Also from Eq. (36) we note that unbiased methods become the preferred option as the expected precision is increased, i.e. when $\varepsilon$ is reduced. We note that there is a threshold value $\varepsilon_{TH} = \frac{27}{4} \frac{|\lambda| C_B}{C_U}$ for which both the unbiased and binomial methods are equally efficient.

Eqs. (33), (34) and (36) conform the main result of this work. These three equations (i) fix the free parameters of the binomial method ($\Delta t$ and $M_B$) in order to compute averages with fixed precision $\varepsilon$ at minimum CPU time usage, and (ii) inform us if the binomial method is more efficient than the unbiased method. The use of these equations require the estimation of four quantities: $\sigma$, $C_U$, $\lambda$, and $C_B$, which can be computed numerically with limited efforts. While $\sigma$ and $\lambda$ rely solely on the process and approximation, hence are expected to remain constant across different machines, both $C_U$ and $C_B$ depend on the machine, but also on the programming language and the user's ability to write efficient codes. The standard deviation $\sigma$ depends only on the random variable $Z$ and has to be computed anyway in order to have a faithful estimate of the errors. As we will show in the examples of section *Numerical study*, the constant $\lambda$ can be obtained through extrapolation at high values of $\Delta t$ (thus, very fast implementations). Finally, the constants $C_U$ and $C_B$ can be determined very accurately and at a little cost by measuring the CPU usage time of a few iterations with standard clock routines. Furthermore, in Supplementary Note 2, we provide a detailed discussion on the estimation of $C_U$ without the need to implement any unbiased method. This approach offers a practical means to determine the value of $C_U$ while avoiding the complexities associated with unbiased methods.

We can also work with alternative rules that fix the relative error, defined to as

$$\varepsilon_r := \left| \frac{\langle Z \rangle - Z_M}{\langle Z \rangle} \right|, \tag{37}$$

instead of the absolute error $\varepsilon$. To do so, we consider that the difference $\langle Z \rangle - Z_M$ is of order $\varepsilon$ and replace $\langle Z \rangle$ by a rough estimation $Z_M$. Then, we can replace in Eqs. (33), (34) and (36)

$$\varepsilon \approx \varepsilon_r |Z_M|, \tag{38}$$

where we note that the errors of using Eq. (38) instead of Eq. (37) are of order $\left( \varepsilon / Z_M \right)^2$. Therefore, working with relative errors result in implicit rules, in the sense that one has to make a rough estimation of the quantity that we aim to estimate (i.e. $\langle Z \rangle$).

In the analysis of errors, the number of agents $N$ plays a crucial role due to its significant impact on the magnitude of fluctuations. For instance, when estimating average densities of individuals, and when the central limit theorem applies, the standard error scales as $\sigma \sim 1/\sqrt{N}$[36]. The average time $\Delta t$ between updates in unbiased methods is expected to be inversely proportional to $N$ (see Supplementary Note 2). Therefore, we expect $C_U \sim N$. Since $\lambda$ is a difference between biased and unbiased estimations, it will have the same scaling with $N$ as the quantity $\langle Z \rangle$ (see Supplementary Note 3). The



**Fig. 3 | Scaling of errors.** Panel **a** plots the average density $\langle x_t \rangle_B := \frac{\langle n_t \rangle_B}{N}$ of infected individuals of the all-to-all SIS model at time $t = 20$ obtained using the binomial method for different values of the discretization step $\Delta t$. The number of realizations is $M_B = 100$, and other parameter values are $\beta = 4$, $\mu = 1$, $N = 10^3$, $n(t = 0) = 10$. The statistical error bars are smaller than the symbol size. In accordance with Eq. (26), we find that the average $\langle x_t \rangle_B$ follows a linear dependence at small $\Delta t$ with slope $\lambda = -0.25(1)$. The horizontal dashed line is the extrapolation at $\Delta t = 0$ of $\langle x \rangle_B$ obtained from the linear fit (continuous line). In panel **b** we plot for the same case, the relative error $\varepsilon_r := |\frac{\langle n_t \rangle_B}{N} - 1|$, using a very accurate value of $\frac{\langle n_t \rangle}{N} = 0.7497$ obtained with the so-called Gaussian approximation[75], corroborating the linear dependence with the discretization step (dashed line of slope $0.25 \times N/\langle n_t \rangle = 0.33$).

constant $C_B$ depends crucially on the method used to sample binomial random variables, and in some cases is independent of $N$, as discussed in Supplementary Note 4. Therefore, when estimating average densities, we anticipate $\alpha$ to decrease with increasing system size, as

$$\alpha \sim 1/N, \tag{39}$$

making the use of biased methods more suitable as the system size grows.

## Numerical study

In this section, we want to compare the performance of the Gillespie algorithm (in representation of the unbiased strategies) and the binomial method (in representation of unbiased synchronous methods). Also, we show the applicability of the rules derived in last section to fix the optimal values of $\Delta t$ and $M_B$, and decide whether the biased or unbiased method is faster. We will do so in the context of the SIS model with all-to-all connections and a more complex SEIR model with meta-population structure.

**All-to-all SIS model.** We study in this section the all-to-all connectivity, where every agent is connected to all others and have the same values of the transition rates. In the particular context of the SIS process, these rates read :

$$w(0 \to 1) = \beta \sum_{j=1}^{N} \frac{s_j}{N} = \beta \frac{n}{N}, \quad w(1 \to 0) = \mu. \tag{40}$$

Where $\mu$ represents the rate at which infected individuals recover from the disease and $\beta$ is the rate at which susceptible individuals contract the disease from an infected contact. The transition rates at the macroscopic description are also easily read from the macroscopic variable itself. From Eq. (5):

$$W(n \to n + 1) = \beta \frac{n}{N} (N - n)$$
$$W(n \to n - 1) = \mu n. \tag{41}$$

The main outcome of this all-to-all setting is well known and can easily be derived from the mean-field equation for the average number of infected individuals[65],

$$\frac{d\langle n(t) \rangle}{dt} = \beta \frac{\langle n \rangle}{N} (N - \langle n \rangle) - \mu \langle n \rangle \tag{42}$$

and indicates that for $R_0 := \beta/\mu > 1$ there is an "active" phase with a non-zero stable steady-sate value $\langle n \rangle_{st} = (1 - \mu/\beta)N$, whereas for $R_0 < 1$ the stable state is the "epidemic-free" phase $\langle n \rangle_{st} = 0$ where the number of infected individuals tends to zero with time.

In order to draw trajectories of this process with the binomial method we use Algorithm 2 with a single class containing all agents, $N_\ell = N$, $n_\ell = n$. The probability to use in the binomial distributions is extracted from the individual rates of Eq. (40):

$$P(1, \Delta t) = 1 - e^{-\mu \Delta t}, \quad P(0, \Delta t) = 1 - e^{-\beta(n/N)\Delta t}. \tag{43}$$

We note that the probability $P(0, \Delta t)$ in Eq. (43) that a susceptible agent experiences a transition in a time $\Delta t$ is an approximation of

$$1 - \exp\left( -\frac{\beta}{N} \int_t^{t+\Delta t} n(t') \, dt' \right). \tag{44}$$

Such approximation is a good representation of the original process when $\Delta t$ is so small that $n(t)$ can be considered as constant in $[t, t + \Delta t]$. In any case, we checked both analytically (see Supplementary Note 3) and numerically [see Fig. 3a, b] that the errors of the method still scale linearly with the time discretization, as pointed out in section The $\frac{27}{4}$ rule.
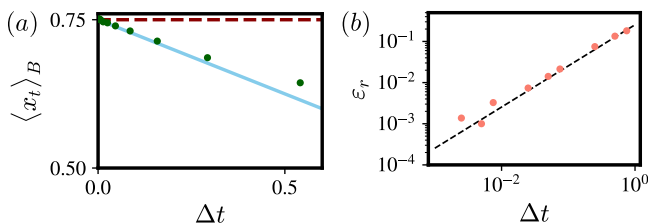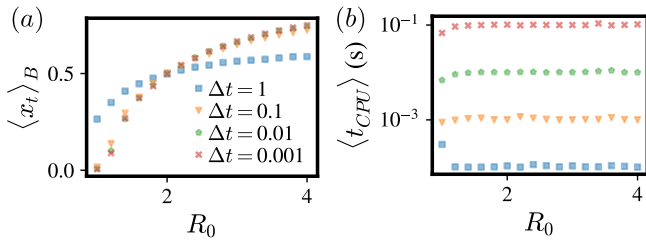
**Fig. 4 | Relevance of time discretization.** We plot in panel **a** the average density $\langle x_t \rangle_B := \frac{\langle n_t \rangle_B}{N}$ of infected individuals of the all-to-all SIS model at time $t = 20$ obtained using the binomial method as a function of $\mathcal{R}_0 = \beta/\mu$ for different discretization times $\Delta t$. We take $n(t=0) = 10$, $\mu = 1.0$, $N = 10^3$, and $M_B = 100$. Statistical error bars are smaller than the symbol size. The estimations of the average agree within errors for $\Delta t = 10^{-3}$ and $\Delta t = 10^{-2}$. However, discrepancies are found for bigger values of $\Delta t$, for which the systematic errors are bigger than the statistical errors. Thus, the analysis of systematic errors should be taken into account to produce results with fixed desired precision. In panel **b**, we plot the average CPU time (in seconds) per realization which, according to Eq. (32) scales as $1/\Delta t$. This figure evidences the need of a fine tuning of $\Delta t$ in order to avoid slow and imprecise calculations.
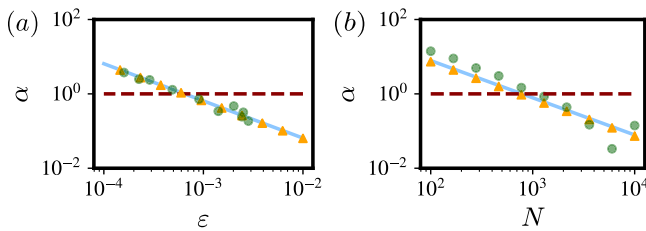


**Fig. 5 | 27/4 rule in all-to-all SIS model.** We plot in panel **a** the ratio between the CPU times of the binomial and the Gillespie algorithms applied to the simulation of an all-to-all SIS model with parameter values $T = 20$, $\mu = 1$, $\beta = 4$, $N = 10^3$, and $n(t=0) = 10$ as a function of the target error $\varepsilon$. The dots are the results of the numerical simulations using the binomial method with the optimal values of the discretization step $\Delta t^{\text{opt}}$ and number of realizations $M_B^{\text{opt}}$ as given by Eqs. (33) and (34), while the number of trajectories in the Gillespie algorithm was computed from Eq. (27). The solid line is Eq. (36), using the values obtained from the simulations: $\lambda = -0.25$, $C_U = 7 \cdot 10^{-3}$ s, $C_B = 2 \cdot 10^{-6}$ s. With triangles we represent results from the use of Eq. (36) with the estimation of $C_U$ explained in Supplementary Note 2. The dashed horizontal line at $\alpha = 1$ signals where the unbiased and biased methods are equally efficient and it crosses the data at $\varepsilon_{TH} = \frac{27}{4} \frac{|\lambda| C_B}{C_U} = 3 \cdot 10^{-3}$. In panel **b** we proceed similar to (a), but fix the precision, and vary the system size $N$. Again, we fix $\Delta t$ and $M_B$ to their optimal values using Eqs. (33) and (34) respectively, and plot results from simulations (dots), our prediction from Eq. (36) measuring $\lambda$, $C_U$, and $C_B$ from simulations (solid line), and Eq. (36) using our theoretical estimation of $C_U$ (triangles) using Eq. B4. This plot is in agreement with the expected scaling of $\alpha$ from Eq. (39). See values of absolute CPU time consumption in Supplementary Note 5.

Now, let us illustrate the relevance of choosing an appropriate discretization $\Delta t$ for the binomial method. First we look for a condition on $\Delta t$ that ensures that Eq. (44) can be properly approximated by Eq. (43). Since the average time between updates at the non-zero fixed point $W(n_{st})^{-1} = [2\mu(1 - \mu/\beta)N]^{-1}$, a heuristic sufficient condition to ensure proper integration is to fix $\Delta t \propto 1/N$. In Fig. 4a, it is shown that this sufficient condition indeed generates a precise integration of the process. Also in Fig. 4a we can see that this is in contrast with the use of $\Delta t = 1$, which provides a poor representation of the process (as claimed in[40]). However, regarding the CPU-time consumption, the sufficient option performs poorly [Fig. 4b]. Therefore, a proper balance between precision and CPU time consumption requires to fine tune the parameter $\Delta t$. This situation highlights the relevance of the rule derived in section The $\frac{27}{4}$ rule to choose $\Delta t$ and discern if the binomial method is advantageous with respect to the unbiased counterparts.

In Fig. 5a, we show the agreement of Eqs. (36) and (39) with results from simulations. In this figure, the discretization step $\Delta t$ and number of

realizations for the binomial method $M_B$ have been optimally chosen according to Eqs. (33) and (34). This figure informs us that the binomial method is more efficient than an unbiased Gillespie algorithm counterpart for a system of size $N = 10^3$ when the target error is large, namely for $\varepsilon > 3 \cdot 10^{-3}$, whereas the unbiased method should be the preferred choice for dealing with high precision estimators. In Fig. 5b we fix the precision and vary the system size $N$ to check that $\alpha$ is inversely proportional to $N$ [Eq. (39)]. Thus, the efficiency of biased methods tends to overcome unbiased approaches as the system size grows. Both in Fig. 5a, b, we show that it is possible to use estimations of $C_U$ without actually having to implement the unbiased method (see Supplementary Note 2). This finding highlights the possibility of achieving accurate results while avoiding the complexities associated with implementing biased methods. It is relevant for the application of the $\frac{27}{4}$ rule that CPU time consumption is not highly dependent on $\mathcal{R}_0$ (as demonstrated in Fig. 3b). Therefore, the efficiency study can be conducted at fixed $\mathcal{R}_0$ values.

**Meta-population SEIR model.** Next, we show that our results hold in a more complex model involving meta-population connectivity and many-state agents. The meta-population framework consist on $\mathcal{C}$ sub-systems or classes, such that class $\ell = 1, \ldots, \mathcal{C}$ contains a population of $N_\ell$ individuals. Agents of different sub-populations are not connected and therefore cannot interact, whereas agents within the same population interact through an all-to-all scheme similar to the one used in section *All-to-all SIS model*. Individuals can diffuse through populations, thus infected individuals can move to foreign populations and susceptible individuals can contract the disease abroad. Diffusion is tuned by a mobility matrix **m**, being the element $m_{\ell,\ell'}$ the rate at which individuals from population $\ell$ travel to population $\ell'$. Therefore, to fully specify the state of agent $i$ we need to give its state $s_i$ and the sub-population $\ell_i$ it belongs to at a given time. Regarding the macroscopic description of the system, the inhabitants of a population can fluctuate and therefore it is needed to keep track of all the numbers $N_\ell$ as well as the occupation numbers $n_\ell$.

In this case we examine the SEIR paradigmatic epidemic model where agents can exist in one of four possible states: susceptible, exposed, infected, or recovered (see e.g.[44]). The exposed and recovered compartments are new additions compared to the SIS model discussed in the previous section. These compartments represent individuals who have been exposed to the disease but are not yet infectious, and individuals who are immune to the disease respectively. The rates of all processes at the sub-population level are:

$$
\begin{aligned}
W_\ell(S_\ell, E_\ell \to S_\ell - 1, E_\ell + 1) &= \beta I_\ell S_\ell / N_\ell, \\
W_\ell(I_\ell, E_\ell \to I_\ell + 1, E_\ell + -1) &= \gamma E_\ell, \\
W_\ell(I_\ell, R_\ell \to I_\ell - 1, R_\ell + 1) &= \mu I_\ell, \\
W(N_\ell, N_{\ell'} \to N_\ell - 1, N_{\ell'} + 1) &= m_{\ell,\ell'} N_\ell,
\end{aligned}
\tag{45}
$$

where $S_\ell$, $E_\ell$, $I_\ell$, and $R_\ell$ denote the number of susceptible, exposed, infected, and recovered individuals in population $\ell$, respectively.

If we assume homogeneous diffusion, the elements of the mobility matrix are $m_{\ell,\ell'} = m$ if there is a connection between subpopulations $\ell$ and $\ell'$ and $m_{\ell,\ell'} = 0$ otherwise. Also if the initial population distribution is homogeneous, $N_\ell(t=0) = N_0$, $\forall \ell$, then the total exit rate reads:

$$
W(\mathbf{s}) = \sum_{\ell=1}^{\mathcal{C}} \left( \beta I_\ell \frac{S_\ell}{N_\ell} + \gamma E_\ell + \mu I_\ell \right) + m \mathcal{C} N_0,
\tag{46}
$$

which can be expressed as a function of the occupation variables $\{S_\ell, E_\ell, I_\ell, N_\ell\}$. In this case, the average time between mobility-events, $[m\mathcal{C}N_0]^{-1}$, is constant and inversely proportional to the total number of agents $\mathcal{C}N_0$. This makes simulating meta-population models with unbiased methods computationally expensive, as a significant portion of CPU time is devoted to simulating mobility events. The binomial method is, therefore,
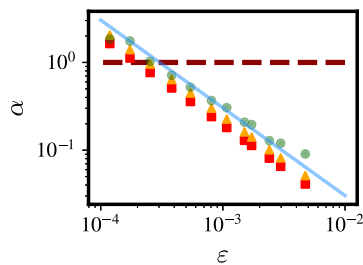
**Fig. 6 | 27/4 rule in meta-population SEIR model.** Similar to Fig. 5 for the case of the meta-population SEIR model with parameter values $t = 7.5$, $\gamma = 1$, $\mu = 1$, $\beta = 4$. There are $\mathcal{C} = 100$ subpopulations arranged in a square $10 \times 10$ lattice such that each subpopulation is connected to 4 nearest neighbors (we assume periodic boundary conditions); each subpopulation contains initially $N_\ell(t = 0) = 10^3$ agents, $\forall\, \ell$. At time zero the state of the system is $I_1(0) = 10$, $I_\ell(0) = 0 \,\forall\, \ell \neq 1$, $E\ell(0) = 0$, $R_\ell(0) = 0$, $S_\ell(0) = N_0 - I_\ell \,\forall\, \ell$. We have set the mobility among neighboring subpopulations to a constant value $m = 10$. The discretization step and the number of trajectories of the binomial method take the optimal values of Eqs. (33) and (34), while the number of trajectories in the Gillespie algorithm was computed from Eq. (27). The required constants measured from the simulations are $\lambda = 0.045$, $C_U = 0.12$ s, $C_B = 1.2 \cdot 10^{-4}$ s. The dashed horizontal line at $\alpha = 1$ signals where the Gillespie and binomial methods are equally efficient and it crosses the data at $\varepsilon_{TH} = \frac{27}{4}\frac{|\lambda| C_B}{C_U} \approx 3 \cdot 10^{-4}$. The continuous line is the theoretical prediction Eq. (36), while circles are results from simulations. Squares and triangles are estimations of $\alpha$ that avoid making simulations of the original process. Squares where obtained through simulations of the all-to-all process. Triangles also use the all-to-all process plus the estimation of $C_U$ using deterministic mean-field equations as outlined in Supplementary Note 2. We note that the values of $\alpha$ are in general agreement across theory, simulations and approximations. See fit for $\lambda$ in (Fig. S5 of the Supplementary Note 5).

the preferred strategy to deal with this kind of process (see Supplementary Note 6 for details on how to apply the binomial method to meta-population models[66]). However, one has to bear in mind that the proper use of the binomial method requires supervising the proper value of $\Delta t$ that generates a faithful description of the process at affordable times.

In Fig. 6 we also check the applicability of the rules derived in section The $\frac{27}{4}$ rule, this time in the context of metapopulation models. As in the case of all-to-all interactions, the preferential use of the binomial method is conditioned to the desired precision for the estimator. Indeed, unbiased methods become more convenient as the target errors decrease.

**Efficient calculation of $C_U$ and $C_B$.** In principle, the use of Eq. (36) requires the implementation of both the unbiased and biased methods to estimate the constants $C_U$ and $C_B$. It would be preferable to devise rules that do not require both implementations, as they can become cumbersome for complex processes with numerous reactions. To address this issue, we propose two approximations to Eq. (36). The first approximation consists of conducting the efficiency analysis on a simpler all-to-all system rather than on the meta-population structure, as outlined in Supplementary Note 2. Our second proposal entirely avoids the implementation of the unbiased method, opting instead for the mean-field estimation of $C_U$ as also described in Supplementary Note 2. In Fig. 6, we also illustrate the concurrence between these two approximations and the direct application of Eq. (36). Overall, Fig. 6 shows the advantage of using the binomial method for low precision. Compared to the case of the all-to-all interactions of section *All-to-all SIS model*, the required CPU-time of the Gillespie method is very large, making it computationally very expensive to use. Therefore, this situation exemplifies the superiority of the binomial method with optimal choices for the discretization times and number of realizations when dealing with complex processes.

**Final implementation.** Summing up, we propose the following steps to use the results of section The $\frac{27}{4}$ rule.

**Algorithm 3**. Rule 27/4

1: Estimate a target quantity, $\langle Z \rangle$, using the biased method with several (large) values of $\Delta t$. Plotting the estimations versus $\Delta t$, compute $\lambda$ as the slope of the linear fit [see Fig. 3a and Fig. S6 of Supplementary Note 5 for examples].

2: Estimate $C_U$ and $C_B$ on simple all-to-all process. Alternatively, estimate $C_U$ using deterministic mean-field calculations as in Supplementary Note 2. Estimations can be done at a small system size $N_s$, then $C_U$ at target system size $N$ is recovered through $C_U(N) = C_U(N_s)N/N_s$.

3: Use Eq. (36) to discern whether the unbiased (for $\alpha > 1$) or biased (for $\alpha < 1$) approach are the most efficient option.

4: If the biased method is the preferred option, then use Eqs. (33) and (34) to fix the discretization time and number of realizations respectively.

## Conclusion

This work provides useful insight into the existing debate regarding the use of the binomial approximation to sample stochastic trajectories. The discretization time of the binomial method needs to be chosen carefully since large values can result in errors beyond the desired precision, while low values can produce extremely inefficient simulations. A proper balance between precision and CPU time consumption is necessary to fully exploit the potential of this approximation and make it useful.

We have demonstrated, through both numerical and analytical evidence, that the systematic errors of the binomial method scale linearly with the discretization time. Using this result, we can establish a rule for selecting the optimal discretization time and number of simulations required to estimate averages with a fixed precision while minimizing CPU time consumption. Furthermore, when comparing specific biased and unbiased implementations, we have derived a rule to identify the more efficient option.

It is not possible to determine whether the unbiased or biased approach is the best option in absolute terms. CPU time consumption varies depending on factors such as the programming language, the machine used for calculations, and the user's coding proficiency. This variability is parametrized through the constants $C_U$ and $C_B$ in our theory. Nevertheless, we can make general statements independent of the implementation. Firstly, the advantage of using the binomial method depends on the target precision: the use of unbiased methods becomes more optimal as the target precision increases. Second, since CPU time scaling with the number of reactions depends on the method, biased methods tend to outperform unbiased methods as the complexity of the model increases.

The numerical study of our proposed rules signals that the ratio of CPU times between the unbiased and binomial methods are similar in both all-to-all and meta-population structures. This result facilitates the use of the rules in the latter case. Indeed, one can develop the study of efficiency in the all-to-all framework and then use the optimal values of the discretization time and number of realizations in the more complex case of meta-populations.

Our work contributes to the generation of trustworthy and fast stochastic simulations, crucial for many real-world applications. Future work will focus on generalizing this approach to deal with adaptive discretizations and address cases involving non-Poissonian processes (see e.g. ref. 67), where unbiased algorithms are challenging to implement.

## Data availability

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

## Code availability

The codes for the different models are available at[68] and are free to use providing the right credit to the author is given.

Different binomial random number generators from different authors were used for robustness analysis[66,69–74]. Check Supplementary Note 4 for details.

## References

1. Britton, T. Stochastic epidemic models: a survey. *Math. Biosci.* **225**, 24–35 (2010).
2. Allen, L. J. An introduction to stochastic epidemic models, in *Mathematical epidemiology* (eds Brauer, F., van den Driessche, P. & Wu, J.) 81–130 (Springer, 2008).
3. Andersson, H. & Britton, T. *Stochastic Epidemic Models and their Statistical Analysis* Vol. 151 (Springer Science & Business Media, 2012).
4. Pastor-Satorras, R., Castellano, C., Van Mieghem, P. & Vespignani, A. Epidemic processes in complex networks. *Rev. Mod. Phys.* **87**, 925 (2015).
5. Brauer, F. Mathematical epidemiology: past, present, and future. *Infectious Dis. Modelling* **2**, 113–127 (2017).
6. Eubank, S. et al. Modelling disease outbreaks in realistic urban social networks. *Nature* **429**, 180–184 (2004).
7. Ferguson, N. M. et al. Strategies for containing an emerging influenza pandemic in Southeast Asia. *Nature* **437**, 209–214 (2005).
8. Longini Jr, I. M. et al. Containing pandemic influenza at the source. *Science* **309**, 1083–1087 (2005).
9. Germann, T. C., Kadau, K., Longini Jr, I. M. & Macken, C. A. Mitigation strategies for pandemic influenza in the united states. *Proc. Natl Acad. Sci.* **103**, 5935–5940 (2006).
10. Ciofi degli Atti, M. L. et al. Mitigation measures for pandemic influenza in Italy: an individual based model considering different scenarios. *PLoS ONE* **3**, e1790 (2008).
11. Merler, S. & Ajelli, M. The role of population heterogeneity and human mobility in the spread of pandemic influenza. *Proc. Roy. Soc. B: Biol. Sci.* **277**, 557–565 (2010).
12. Aleta, A. et al. Modelling the impact of testing, contact tracing and household quarantine on second waves of COVID-19. *Nature Hum. Behaviour* **4**, 964–971 (2020).
13. Sattenspiel, L. & Dietz, K. A structured epidemic model incorporating geographic mobility among regions. *Math. Biosci.* **128**, 71–91 (1995).
14. Colizza, V., Barrat, A., Barthélemy, M. & Vespignani, A. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. Natl Acad. Sci.* **103**, 2015–2020 (2006).
15. Colizza, V., Barrat, A., Barthelemy, M., Valleron, A.-J. & Vespignani, A. Modeling the worldwide spread of pandemic influenza: baseline case and containment interventions. *PLOS Med.* **4** (2007).
16. Balcan, D. et al. M. *Proc. Natl Acad. Sci.* **106**, 21484–21489 (2009).
17. Balcan, D. et al. Modeling the spatial spread of infectious diseases: The GLobal Epidemic and Mobility computational model. *J. Comput. Sci.* **1**, 132–145 (2010).
18. Balcan, D. & Vespignani, A. Phase transitions in contagion processes mediated by recurrent mobility patterns. *Nat. Phys.* **7**, 581–586 (2011).
19. Merler, S. et al. Spatiotemporal spread of the 2014 outbreak of Ebola virus disease in Liberia and the effectiveness of non-pharmaceutical interventions: a computational modeling analysis. *Lancet Infectious Dis.* **15**, 204–211 (2015).
20. Zhang, Q. et al. Spread of Zika virus in the Americas. *Proc. Natl Acad. Sci. USA* **114**, E4334–E4343 (2017).
21. Gómez-Gardeñes, J., Soriano-Panos, D. & Arenas, A. Critical regimes driven by recurrent mobility patterns of reaction–diffusion processes in networks. *Nat. Phys.* **14**, 391–395 (2018).
22. Gilbert, M. et al. Preparedness and vulnerability of African countries against importations of COVID-19: a modeling study. *Lancet* **395**, 871–877 (2020).
23. Chinazzi, M. et al. The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. *Science* **368**, 395–400 (2020).
24. Arenas, A. et al. Modeling the spatiotemporal epidemic spreading of COVID-19 and the impact of mobility and social distancing interventions. *Phys. Rev. X* **10**, 041055 (2020).
25. Aguilar, J. et al. Impact of urban structure on infectious disease spreading. *Sci. Rep.* **12**, 1–13 (2022).
26. Argun, A., Callegari, A. & Volpe, G. *Simulation of Complex Systems* (IOP Publishing, 2021).
27. Thurner, S., Hanel, R. & Klimek, P. *Introduction to the Theory of Complex Systems* (Oxford University Press, 2018).
28. Tranquillo, J. V. *An Introduction to Complex Systems* (Springer, 2019).
29. Odum, E. P. & Barrett, G. W. *Fundamentals of Ecology* Vol. 3 (Saunders Philadelphia, 1971).
30. Balaban, N. Q., Merrin, J., Chait, R., Kowalik, L. & Leibler, S. Bacterial persistence as a phenotypic switch. *Science* **305**, 1622–1625 (2004).
31. Elowitz, M. B., Levine, A. J., Siggia, E. D. & Swain, P. S. Stochastic gene expression in a single cell. *Science* **297**, 1183–1186 (2002).
32. Kiviet, D. J. et al. Stochasticity of metabolism and growth at the single-cell level. *Nature* **514**, 376–379 (2014).
33. Rolski, T., Schmidli, H., Schmidt, V. & Teugels, J. L. *Stochastic Processes for Insurance and Finance* (John Wiley & Sons, 2009).
34. Sornette, D. Critical market crashes. *Phys. Rep.* **378**, 1–98 (2003).
35. Baccelli, F. & Błaszczyszyn, B. et al. Stochastic geometry and wireless networks: Volume II Applications. *Foundations Trends Networking* **4**, 1–312 (2010).
36. van Kampen, N. *Stochastic Processes in Physics and Chemistry* (Elsevier Science Publishers, 1992).
37. Milz, S. & Modi, K. Quantum stochastic processes and quantum non-Markovian phenomena. *PRX Quantum* **2**, 030201 (2021).
38. Ramaswamy, S. The mechanics and statistics of active matter. *Annu. Rev. Condens. Matter Phys.* **1**, 323–345 (2010).
39. Gillespie, D. T. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.* **58**, 35–55 (2007).
40. Fennell, P. G., Melnik, S. & Gleeson, J. P. Limitations of discrete-time approaches to continuous-time contagion dynamics. *Physi. Rev. E* **94**, 052125 (2016).
41. Gómez, S., Gómez-Gardenes, J., Moreno, Y. & Arenas, A. Nonperturbative heterogeneous mean-field approach to epidemic spreading in complex networks. *Physical Review E* **84**, 036105 (2011).
42. Leff, P. The two-state model of receptor activation. *Trends Pharmacol. Sci.* **16**, 89–97 (1995).
43. Brush, S. G. History of the Lenz-Ising model. *Rev. Mod. Phys.* **39**, 883 (1967).
44. Keeling, M. J. & Rohani, P. *Modeling Infectious Diseases in Humans and Animals* (Princeton University Press, 2011).
45. Fernández-Gracia, J., Suchecki, K., Ramasco, J. J., San Miguel, M. & Eguíluz, V. M. Is the voter model a model for voters? *Phys. Rev. Lett.* **112**, 158701 (2014).
46. Lee, B. & Graziano, G. A two-state model of hydrophobic hydration that produces compensating enthalpy and entropy changes. *J. Am. Chem. Soc.* **118**, 5163–5168 (1996).
47. Huang, H. W. Action of antimicrobial peptides: two-state model. *Biochemistry* **39**, 8347–8352 (2000).
48. Bridges, T. M. & Lindsley, C. W. G-protein-coupled receptors: from classical modes of modulation to allosteric mechanisms. *ACS Chem. Biol.* **3**, 530–541 (2008).
49. Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* **22**, 403–434 (1976).
50. Toral, R. & Colet, P. *Stochastic Numerical Methods: an Introduction for Students and Scientists* (John Wiley & Sons, 2014).
51. Cota, W. & Ferreira, S. C. Optimized Gillespie algorithms for the simulation of Markovian epidemic processes on large and heterogeneous networks. *Comput. Phys. Commun.* **219**, 303–312 (2017).

52. Colizza, V. & Vespignani, A. Epidemic modeling in metapopulation systems with heterogeneous coupling pattern: theory and simulations. *J. Theoret. Biol.* **251**, 450–467 (2008).

53. Tailleur, J. & Lecomte, V. Simulation of large deviation functions using population dynamics. *AIP Conf. Proc.* Vol. 1091, 212–219 (2009).

54. Masuda, N. & Vestergaard, C. L. Gillespie algorithms for stochastic multiagent dynamics in populations and networks. In *Elements in Structure and Dynamics of Complex Networks*, (ed Guido Caldarelli) (Cambridge University Press, 2022).

55. Colizza, V., Pastor-Satorras, R. & Vespignani, A. Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nat. Phys.* **3**, 276–282 (2007).

56. Goutsias, J. & Jenkinson, G. Markovian dynamics on complex reaction networks. *Phys. Rep.* **529**, 199–264 (2013).

57. Allen, L. J. Some discrete-time SI, SIR, and SIS epidemic models. *Math. Biosci.* **124**, 83–105 (1994).

58. Leier, A., Marquez-Lago, T. T. & Burrage, K. Generalized binomial $\tau$-leap method for biochemical kinetics incorporating both delay and intrinsic noise. *J. Chem. Phys.* **128** (2008).

59. Peng, X., Zhou, W. & Wang, Y. Efficient binomial leap method for simulating chemical kinetics. *J. Chem. Phys.* **126** (2007).

60. Cao, Y., Gillespie, D. T. & Petzold, L. R. Efficient step size selection for the tau-leaping simulation method. *J. Chem. Phys.* **124** (2006).

61. Cao, Y. & Samuels, D. C. Discrete stochastic simulation methods for chemically reacting systems. *Methods Enzymol.* **454**, 115–140 (2009).

62. Lecca, P. Stochastic chemical kinetics: a review of the modelling and simulation approaches. *Biophys. Rev.* **5**, 323–345 (2013).

63. Rathinam, M., Petzold, L. R., Cao, Y. & Gillespie, D. T. Consistency and stability of tau-leaping schemes for chemical reaction systems. *Multiscale Modeling Simul.* **4**, 867–895 (2005).

64. Asmussen, S. & Glynn, P. W. *Stochastic Simulation: Algorithms and Analysis* Vol. 57 (Springer, 2007).

65. Marro, J. & Dickman, R. *Nonequilibrium Phase Transitions in Lattice Models. Collection Alea-Saclay: Monographs and Texts in Statistical Physics* (Cambridge University Press, 1999).

66. Chandler, R. & Northrop, P. The FORTRAN library RANDGEN http://www.homepages.ucl.ac.uk/~ucakarc/work/software/randgen.f. See documentation at https://www.ucl.ac.uk/~ucakarc/work/software/randgen.txt.

67. Ferguson, N. M. et al. Strategies for mitigating an influenza pandemic. *Nature* **442**, 448–452 (2006).

68. Aguilar, J. GitHub repository. https://github.com/jvrglr/Biased-versus-unbiased-codes.git.

69. The FORTRAN library ranlib.f. https://www.netlib.org/random/.

70. Python's numpy.random.binomial source code, as C function random_binomial: https://fossies.org/linux/numpy/numpy/random/src/distributions/distributions.c.

71. Kachitvichyanukul, V. & Schmeiser, B. W. Binomial random variate generation. *Commun. ACM* **31**, 216–222 (1988).

72. Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. *Numerical Recipes FORTRAN 90: the Art of Scientific and Parallel Computing* (Cambridge University Press, 1996).

73. Davis, C. S. The computer generation of multinomial random variates. *Comput. Statistics Data Anal.* **16**, 205–217 (1993).

74. Fishman, G. S. Sampling from the binomial distribution on a computer. *J. Am. Statistical Assoc.* **74**, 418–423 (1979).

75. Lafuerza, L. F. & Toral, R. On the Gaussian approximation for master equations. *J. Statistical Phys.* **140**, 917–933 (2010).

## Author contributions

J.A., J.J.R., and R.T. conceived and designed the study. J.A. performed the simulations. J.A., J.J.R., and R.T. wrote the paper. All the authors read and approved the paper.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s42005-024-01648-z.

**Correspondence** and requests for materials should be addressed to Javier Aguilar.

**Peer review information** *Communications Physics* thanks the anonymous reviewers for their contribution to the peer review of this work

**Reprints and permissions information** is available at http://www.nature.com/reprints

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# SUPPLEMENTARY INFORMATION
## Biased versus unbiased numerical methods for stochastic simulations: application to contagion processes

Javier Aguilar, José J. Ramasco, and Raúl Toral.
Instituto de Física Interdisciplinar y Sistemas Complejos IFISC (CSIC-UIB), Campus UIB, 07122 Palma de Mallorca, Spain.
Departamento de Electromagnetismo y Física de la Materia e Instituto Carlos I de Física Teórica y Computacional, Universidad de Granada, Granada E-18071, Spain .

---

### Supplementary Note 1.   OPTIMAL TIME

In this section, we derive expressions equivalent to Eqs. (33), (34), and  (36) in the main text but for arbitrary scaling relations.  First, we consider the situation in which the systematic errors of the biased method scale as an arbitrary power of the discretization time,

$$\langle Z \rangle = Z_{M_B} + \lambda \left(\Delta t\right)^\varphi \pm \varepsilon_B. \tag{S1}$$

Considering this generalization is pertinent since sub-linear and super-linear scalings ($\varphi < 1$ and $\varphi > 1$ respectively) can occur within the context of numerical methods for simulating stochastic models, for example, in the context of the numerical integration of stochastic differential equations [51, 65]. We also consider the case in which the scaling of the CPU time with the number of realizations can be sublinear,

$$t_B^{(\text{CPU})} = C_B \left(M_B\right)^\psi \frac{T}{\Delta t}, \tag{S2}$$

and

$$t_U^{(\text{CPU})} = C_U M^\psi T. \tag{S3}$$

Non-trivial values for $\psi$ could be obtained when working on massively parallel architectures like GPUs.

Since error sources can only accumulate, the only way to reduce errors in biased computations is to increase the number of realizations ($M_B$). The errors using a biased method equals the ones of the unbiased counterpart when

$$M_B = M \left( \frac{|\lambda| \left(\Delta t\right)^\varphi}{\varepsilon} - 1 \right)^{-2}. \tag{S4}$$

Inserting Eq. (S4) in Eq. (S2) we obtain:

$$t_B^{(\text{CPU})} = \frac{M C_B T}{\Delta t} \left( \frac{|\lambda| \left(\Delta t\right)^\varphi}{\varepsilon} - 1 \right)^{-2\psi}. \tag{S5}$$

The above equation informs about the CPU time consumption using the binomial method with a time discretization $\Delta t$ with general scaling relations. Eq.(S5) has two relative extrema. We discard one of them, $(\Delta t)^\varphi = \varepsilon/|\lambda|$, since its use would require sampling infinite-many biased trajectories [see Eq. (S4)]. The other relative extrema is a minimum that we identify as the optimal value for $\Delta t$:

$$\Delta t^{\text{opt}} = \left[ \frac{1}{1 + 2\varphi\psi} \frac{\varepsilon}{|\lambda|} \right]^{\frac{1}{\varphi}}. \tag{S6}$$

Inserting this result in Eq. (S4) we obtain the general expression for the optimal number of realizations to be sampled with the biased method

$$M_B = M \left( 1 + \frac{1}{2\varphi\psi} \right)^2. \tag{S7}$$

Lastly, the general ratio of CPU times reads

$$\alpha = \frac{t_B^{(\text{CPU,opt})}}{t_U^{(\text{CPU})}} = (2\varphi\psi)^{-2\psi} \left(1 + 2\varphi\psi\right)^{2\psi + \frac{1}{\varphi}} \frac{C_B}{C_U} \left(\frac{|\lambda|}{\varepsilon}\right)^{\frac{1}{\varphi}}. \tag{S8}$$

Setting $\psi = \varphi = 1$, Eqs. (S6), (S7), and (S8) reduce to Eqs. (33), (34), and (36) respectively.

## Supplementary Note 2.   ESTIMATION AND SCALING OF THE CONSTANT C

In the main text, we use the constant $C_U$ to characterize the CPU time usage of unbiased methods,

$$t_U^{(\text{CPU})} = C_U M T. \tag{S9}$$

Where $M$ is the number of realizations and $T$ is the simulation time at which the program stops. The constant $C_U$ is employed in Eq.(36) to determine the relative efficiency between the biased and unbiased methods for computing averages with fixed precision. While it is possible to measure $C_U$ with limited computational efforts, it still requires the implementation of an unbiased algorithm. In this section, we propose estimations of $C_U$ that circumvent such implementation, simplifying the use of Eq.(36).

### A.   Deterministic mean-field approximation

For the first approximation, we rewrite Eq. (S9) in terms of the average CPU time to execute one iteration with the unbiased method ($t_{U,\text{it}}^{(\text{CPU})}$) and the average number of iterations required to simulate one trajectory of duration $T$ ($\mathcal{N}_T$),

$$t_U^{(\text{CPU})} = t_{U,\text{it}}^{(\text{CPU})} \mathcal{N}_T M. \tag{S10}$$

To proceed with the theoretical estimation of $\mathcal{N}_T$, we rewrite it as

$$\mathcal{N}_T = \frac{T}{\langle \Delta t \rangle_T}, \tag{S11}$$

where $\langle \Delta t \rangle_T$ is the average time between updates of the process along a trajectory of duration $T$. Hence, we can estimate the constant $C_U$ as

$$C_U \approx \frac{t_{U,\text{it}}^{(\text{CPU})}}{\langle \Delta t \rangle_T}. \tag{S12}$$

The advantage of using Eq. (S12) comes from the fact that we can estimate $t_{U,\text{it}}^{(\text{CPU})}$ without implementing the unbiased method in a program. Instead, we can measure it as the average CPU time consumption that it takes to execute the operations involved in one iteration of the exact algorithm. Furthermore, we propose to estimate $\langle \Delta t \rangle_T$ using the deterministic mean field approximation of the process,

$$\langle \Delta t \rangle_T = \frac{1}{T} \int_0^T \frac{1}{W[n(t)]} dt. \tag{S13}$$

Where $W[n(t)]$ is the total exit rate defined in Eq. (11) evaluated in the state of the system at time $t$, and using the deterministic mean-field dynamics. For example, in the case of the all-to-all SIS model,

$$W[n(t)] = n(t) \left( \beta \frac{N - n(t)}{N} + \mu \right), \tag{S14}$$

where $n(t) = Nx(t)$, and $x(t)$ is the solution to

$$\frac{d}{dt}x(t) = x(t)\left(\beta(1 - x(t)) - \mu\right),$$ (S15)

which is

$$x(t) = \frac{e^{(\beta - \mu)t}}{\frac{R_0\left(e^{(\beta - \mu)t} - 1\right)}{R_0 - 1} + \frac{1}{x_0}},$$ (S16)

The integral in Eq. S13 for the SIS model reads

$$\langle \Delta t \rangle_T = \frac{R_0}{2\mu(R_0 - 1)} - \frac{\left(1 - e^{(1 - R_0)\mu T}\right)(R_0(x_0 - 1) + 1)}{(R_0 - 1)^2(R_0 + 1)\mu T x_0}$$
$$- \frac{R_0}{2(R_0 + 1)^2\mu T}\log\left[\frac{(R_0 - 1)(R_0(x_0 - 1) - 1)}{(R_0 + 1)e^{T - R_0\mu T}(R_0(x_0 - 1) + 1) - 2R_0 x_0}\right],$$ (S17)

with

$$R_0 = \frac{\beta}{\mu}.$$ (S18)

For the particular case of systems with stable or meta-stable states, like the SIS all-to-all model, one can further approximate this derivation and evaluate Eq. (S13) in the non-zero stable state ($n_{\text{st}}$).

$$\lim_{T \to \infty}\langle \Delta t \rangle_T = \frac{1}{W[n_{\text{st}}]} = \frac{R_0}{2\mu(R_0 - 1)}.$$ (S19)

In Fig. S1, we show the agreement between Eqs. (S17) and (S19) and results from simulations.
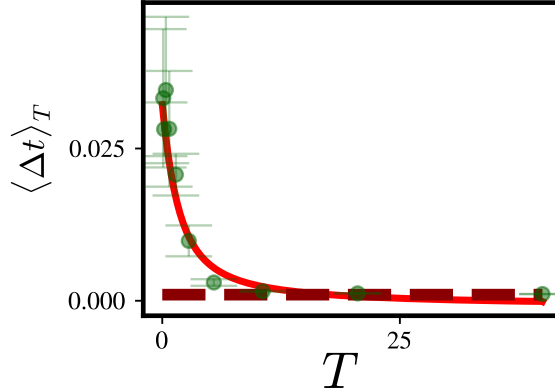


FIG. S1. Estimation of $\langle \Delta t \rangle_T$ for different values of the final time $T$ using the mean-field formula from Eq. (S17) (solid line), its long-time limit (Eq. (S19) in dashed line), and measures from simulations (dots and errorbars noting the 97.5$^{\text{th}}$ porcentile). Parameters: $N = 1000$, $\beta = 2$, $\mu = 1$.

In order to produce the dotted line in Fig. 5-(a), we used Eqs. (S12) and (S17), where we measured $t_{U,\text{it}}^{(\text{CPU})}$ as the time needed to generate one uniform random number, make its logarithm, execute an *if*- statement, and carry the arithmetic operations that it would require to compute the exit rate.

### 1. Scaling with system size

As we can see through this example, it is generally true that the total exit rate is extensive in the number of agents, this is, we can express the exit rate as

$$W\left[n(t)\right] = Nf\left[x(t)\right],\tag{S20}$$

where $f\left[x(t)\right]$ is an intensive function. Putting all together, the CPU time with the unbiased method reads

$$t_U^{(\mathrm{CPU})} = \frac{t_{U,\mathrm{it}}^{(\mathrm{CPU})}TN}{\int_0^T f^{-1}(t)dt}MT,\tag{S21}$$

where we can identify an estimation of $C_U$,

$$C_U = \frac{t_{U,\mathrm{it}}^{(\mathrm{CPU})}TN}{\int_0^T f^{-1}(t)dt}.\tag{S22}$$

Therefore, we expect that $C_U$ scales, at least, linearly with the system size. Depending on the process, the scaling could be super-linear, since $t_{U,\mathrm{it}}^{(\mathrm{CPU})}$ depends on the number of reactions, which could depend itself on the number of agents. Thus, it is possible to do the efficiency study on small systems (of size $N_s << N$) and then scale to the values of $C_U$ at big $N$ with

$$C_U(N) = C_U(N_s)N/N_s\tag{S23}$$

### 2. SEIR model

Eq. (S12) can also be used to approximate the constant $C_U$ for multidimensional models, like the SEIR model of section *Meta-population SEIR model*. The deterministic mean-field equations of the SEIR model read

$$\begin{cases}\frac{d}{dt}s(t) = -\beta s(t)i(t),\\ \frac{d}{dt}e(t) = \beta s(t)i(t) - \gamma e(t),\\ \frac{d}{dt}i(t) = \gamma e(t) - \mu i(t),\\ \frac{d}{dt}r(t) = \mu i(t).\end{cases}\tag{S24}$$

With $s := S/N$, $e := E/N$, $i := I/N$, $r := R/N$, and $N = S + E + I + R$. One more time, we can estimate the average time between updates with the integral

$$\langle\Delta t\rangle_T = \frac{1}{T}\int_0^T \frac{1}{W\left[\boldsymbol{s}(t)\right]}dt.\tag{S25}$$

Where the total exit rate for the all-to-all SEIR model reads

$$W\left[\boldsymbol{s}(t)\right] = W\left[S, E, I, R\right] = \beta I\frac{S}{N} + \gamma E + \mu I.\tag{S26}$$

The triangles in Fig. 6 where obtained integrating Eqs. (S24) and (S25) numerically with $N = 1000$, $\beta = 4$, $\gamma = 1, \mu = 1$, $S(t = 0) = N - 10$, $I(t = 0) = 10$, $E(t = 0) = R(t = 0)$, $T = 2.4$.

## B. All-to-all approximation

The second approach that we propose can be applied to meta-population models. The idea is to assess efficiency, as outlined in section *The $\frac{27}{4}$ rule*, but based on a simplified all-to-all model with $N$ agents. Thus ignoring the meta-population structure. For the case of the SEIR model described in section *Meta-population SEIR model*, the rates for

its associated all-to-all version read,

$$
\begin{aligned}
W(S, E \to S - 1, E + 1) &= \beta\, I\, S/N, \\
W(I, E \to I + 1, E + -1) &= \gamma\, E, \\
W(I, R \to I - 1, R + 1) &= \mu\, I, \\
N &= S + E + I + R.
\end{aligned} \tag{S27}
$$

Let $C_U^{\text{ATA}}$ and $C_B^{\text{ATA}}$ be the constants appearing in Eq. (36) for the all-to-all setting, then we approximate the constants $C_U$ and $C_B$ associated to the meta-population system as

$$
C_B \approx C_B^{\text{ATA}} \quad C_U \approx C_U^{\text{ATA}} \frac{N}{N_0\, \mathcal{C_U}}, \tag{S28}
$$

where we assume the scalings with the system size for $C_U$ and $C_B$ discussed respectively in Supplementary Note 2, and Supplementary Note 4. In this way, one substitutes the implementations on a meta-population system with $\mathcal{C_U} \cdot N_0$ agents with much simpler all-to-all implementations with $N$ agents.

**Supplementary Note 3. ESTIMATION AND SCALING WITH SYSTEM SIZE OF THE CONSTANT $\lambda$**

Consider a SIS model with all-to-all interactions, and let $n(t)$ be the number of infected individuals at time $t$. The probability that a susceptible agent will change its state at time $t' \in [t, t + \Delta t]$ is:

$$
P(0, \Delta t)_{\text{exact}} = 1 - \exp\left(-\frac{\beta}{N} \int_t^{t + \Delta t} n(s)ds\right). \tag{S29}
$$

In the context of the binomial approximation, this probability is approximated by:

$$
P(0, \Delta t) = 1 - \exp\left(-\frac{\beta}{N} n(t)\Delta t\right). \tag{S30}
$$

The difference between Eqs. (S29) and (S30) is the error associated to the use of Eq. (S30) instead of Eq.(S29). We call this difference $\Delta P$.

$$
\Delta P = P(0, \Delta t)_{\text{exact}} - P(0, \Delta t) = \exp\left(-\frac{\beta}{N} n(t)\Delta t\right) - \exp\left(-\frac{\beta}{N} \int_t^{t + \Delta t} n(s)ds\right). \tag{S31}
$$

Considering $\Delta t$ small, we can approximate

$$
\int_t^{t + \Delta t} n(s)ds \approx n(t)\Delta t + \dot{n}(t)\frac{\Delta t^2}{2}. \tag{S32}
$$

Where $\dot{n}(t) = \frac{d}{dt}n(t)$. Inserting the above expression in Eq. (S31), we obtain

$$
\begin{aligned}
\Delta P &= \exp\left(-\frac{\beta}{N} n(t)\Delta t\right) - \exp\left(-\frac{\beta}{N}\left[n(t)\Delta t + \dot{n}(t)\frac{\Delta t^2}{2}\right]\right) \\
&= \exp\left(-\frac{\beta}{N} n(t)\Delta t\right)\left[1 - \exp\left(-\frac{\beta}{N}\dot{n}(t)\frac{\Delta t^2}{2}\right)\right] \approx \frac{\beta\dot{n}(t)}{2N}\Delta t^2.
\end{aligned} \tag{S33}
$$

If we make use of the binomial method, the faithful increment in the number of infected individuals should be $\Delta n_{\text{exact}}$, a random variable drawn from a binomial distribution $\mathbf{B}\left(n(t), P(0, \Delta t)_{\text{exact}}\right)$. Instead, we use a random variable $\Delta n$ drawn from the approximate distribution $\mathbf{B}\left(n(t), P(0, \Delta t)\right)$. The difference between the mean values of the exact random variable and the actual one used in the numerical method is

$$
\langle \Delta n \rangle_{\text{exact}} - \langle \Delta n \rangle = n(t)\Delta P \sim \Delta t^2. \tag{S34}
$$

If we aim to reach a final simulation time $T$, the accumulated error of using the approximation Eq. (S31) for a number of iterations proportional to $T/\Delta t$ scales as $\Delta P/\Delta t \sim \Delta t$. This scaling is corroborated numerically in Fig. 3 of the main text.

Therefore, the estimation for $\lambda$ associated to the number of infected individuals scales linearly with $N$,

$$\lambda \sim \mathcal{O}(N), \tag{S35}$$

whereas the errors over densities $[x(t)]$ are independent of $N$,

$$\lambda \sim \mathcal{O}(1), \tag{S36}$$

## Supplementary Note 4. SCALING OF THE CONSTANT $C_B$

The scaling of the constant $C_B$ with the parameters of the process depends crucially on the algorithm used to generate binomial samples, i.e. random numbers $\mathbf{B}(N, p)$ from a binomial distribution [Eq.(14)] of parameters $N$ and $p$, representing the number of trials and the probability of success, respectively. There are several algorithms to generate $\mathbf{B}(N, p)$ numbers and which one is more convenient (i.e. faster, using less CPU execution time) depends on the specific values of the parameters $N$ and $p$ and, to a lesser extent, on the machine and programming language used. Generally speaking, the methods split in two families: those whose CPU time grows linearly on $N$ and those whose CPU time is independent of $N$. Amongst the first, we mention the numerical inversion of the cumulative distribution function, and the iteration of $N$ Bernouilli processes each one with success probability $p$. Amongst the second set of methods, the simplest one is a rejection algorithm in which a value selected from a Lorenztian distribution is accepted with a carefully chosen probability. A more sophisticated inversion method whose execution time is also independent of $N$ is discussed in Ref. [73]. Other methods whose execution time is independent of the system size exploit the relation between binomial and beta distributions (see e.g. [76]).

One tries to profit from the best of each family by choosing methods whose time scale linearly with $N$ up to a threshold number of trials in which it is more profitable to switch to an $N$-independent algorithm for sufficiently large $N$. In general, the criterion when to choose from one method to another is determined by a threshold value $\Theta_{\mathrm{TH}}$ of the product $Np$ (the expected value of the binomial random variable). Furthermore, as a significant fraction of the time needed for the binomial method of Algorithm 2 is spent in the generation of the random numbers, the above strategy translates into an approximately linear dependence of the time $C_B$, which depends on the discretization time $\Delta t$ through Eq. (8) and on the number of individuals $N$ through the dependence of the occupation numbers $n_\ell$, up to the threshold and constant afterwards,

$$C_B(N, \Delta t) = \begin{cases} c_1 N & \text{if } Np(N, \Delta t) < \Theta_{\mathrm{TH}}. \\ c_2 & \text{otherwise}, \end{cases} \tag{S37}$$

where $c_1$ and $c_2$ are constant numbers depending on the generation method used.

We have considered three different random number generator for the binomial distribution:
(i) The first one is the built-in Python function `numpy.random.binomial` [72] that uses the method based on the inverse of the cumulative function up to $\Theta_{\mathrm{TH}} = 30$, and the sophisticated algorithm discussed in [73] otherwise. We have used this generator in all the simulations of the paper.
(ii) The second is the Fortran routine `ZBQLBIN` available as part of the `randgen` package at [68]. This method is based on sampling from a beta distribution which can be related to the target binomial [76].
(iii) Based on the rejection method implemented in the Fortran routine `BNLDEV` of Ref. [74] and our own timing tests we have written a Fortran routine `iran_bin` which uses the following scheme [70]:

- If $p < 0.15$ and $pN < 15$ use inversion of the cumulative distribution.

- if $p > 0.15$ and $N < 100$ use repetition of Bernouilli processes.

- In all other cases use the routine `BNLDEV` based on rejection.

In Fig. S2 we plot the value of $\alpha$ defined in Eq. (35) for each of the three generators applied to the metapopulation SEIR model described in section *Meta-population SEIR model* with the same parameters as used in Fig. 6. Defining the threshold error value $\epsilon_{\mathrm{TH}}$ as the minimum value for which the binomial method is more efficient than the Gillespie

algorithm, i.e. the one for which $\alpha = 1$, we find that $\epsilon_{\text{TH}}$[generator (iii)] $< \epsilon_{\text{TH}}$ [generator (ii)] $< \epsilon_{\text{TH}}$ [generator (ii)], showing that generator (iii) is the most efficient one. In the same figure, we also plot the theoretical line associated to Eq. (36), which, overall offers a good approximation to the numerical values. Some discrepancies can be attributed to the dependence of $C_B$ on the error $\varepsilon$. To clarify this point, we plot in Fig. S3 the values of $C_B$ estimated from the same simulations used in Fig. S2 as

$$C_B = \frac{\Delta t^{\text{opt}}}{T} \frac{t_B^{(\text{CPU})}}{M_B^{\text{opt}}}, \tag{S38}$$

where $t_B^{(\text{CPU})}$ is the time needed to do the biased simulations used in Fig. S2.

In Fig. S4 we show the absolute times used to generate dots in Fig. S2 with both the biased and unbiased methods. This figure illustrates that all methods operate within similar temporal scales.
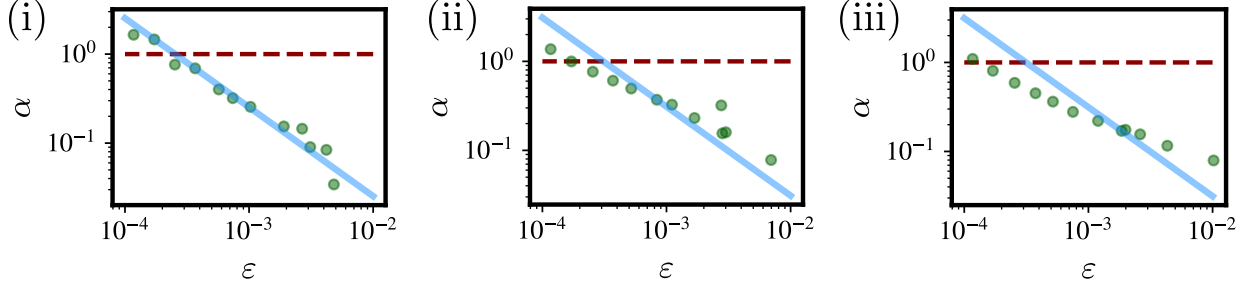


FIG. S2. The figure shows the values of $\alpha$ [Eq. (36)] for different target precisions $\varepsilon$, and the different algorithms explained in the text to extract binomial samples (Methods i), ii), and iii)). The process is the metapopulation SEIR model described in section *Meta-population SEIR model* with the same parameters as used in Fig. 6. Dots are the results from simulation while the continuous line is the prediction of our theory.
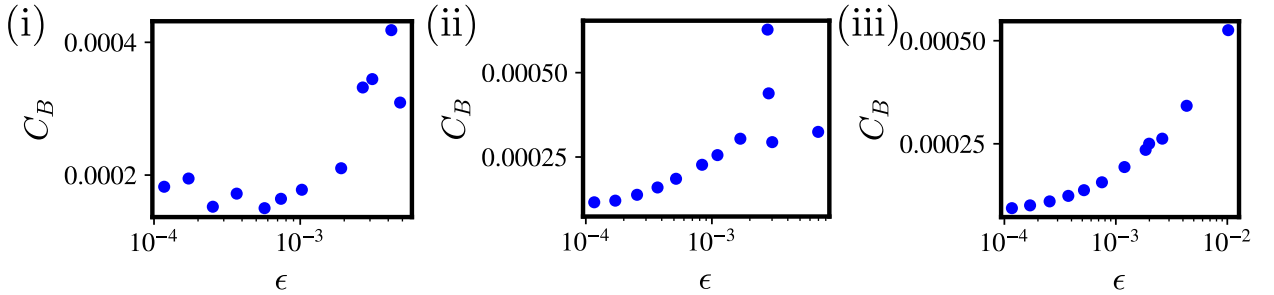


FIG. S3. The figure shows the values of $C_B$ measured from simulations corresponding to dots in Fig. S2.

## Supplementary Note 5.    AUXILIARY FIGURES

### A.    Absolute CPU time consumption

In this section, we present the absolute time counterparts of Fig. 5 instead of its ratio, denoted as $\alpha$.
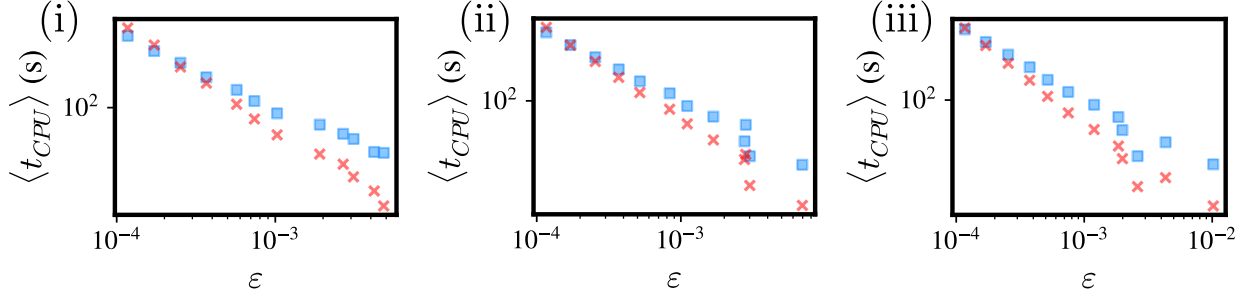
### B.    Scaling of errors SEIR

FIG. S4. The figure shows the absolute times measured from simulations corresponding to dots in Fig. S2 for both the unbiased method (squares) and the biased method (x).
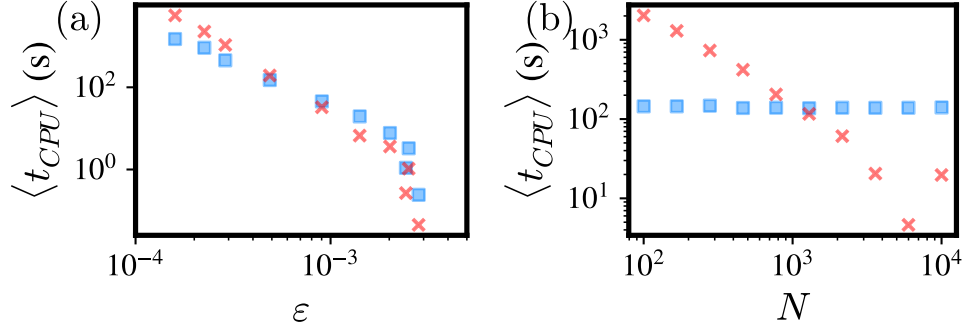


FIG. S5. Average CPU time consumption using the binomial and Gillespie methods (x and squares respectively). The tasks for $(a)$ and $(b)$ correspond to those of Figs. 5 $(a)$ and $(b)$ respectively.
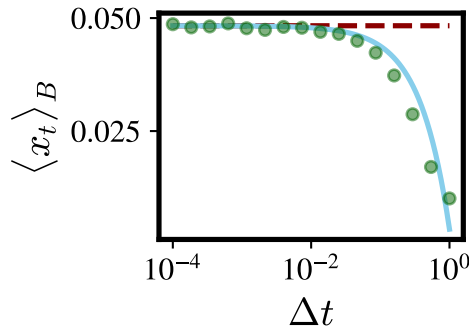


FIG. S6. Average density $\langle x_t \rangle_B := \frac{\langle I_t \rangle_B}{N_0 M}$ of infected individuals of the metapopulation SEIR model at time $t = 7.5$ obtained using the binomial method for different values of the discretization step $\Delta t$. The number of realizations is $M_B = 100$, and other parameter values are the same of Fig. 6. Circles are the results from simulations and the continuous line is a linear fit whose slope is $\lambda = -0.045(1)$. The horizontal dashed line is the extrapolation at $\Delta t = 0$ of $\langle x \rangle_B$ obtained from the linear fit.

## Supplementary Note 6.  BINOMIAL METHOD ON META-POPULATION FRAMEWORK

In this section, we show how to adapt the binomial method (algorithm 2) to the case of meta-population models with SEIR dynamics (described in section *Meta-population SEIR model*). Let $S_\ell(t)$, $E_\ell(t)$, $I_\ell(t)$, and $R_\ell(t)$ be, respectively, the number of susceptible, exposed, infected, and recovered individuals in subpopulation $\ell = 1, \ldots, \mathcal{C}$ at time $t$. These occupation numbers fully characterize the state of the system. Note that the total number of agents in class $\ell$ at time $t$ is $N_\ell(t) = S_\ell(t) + E_\ell(t) + I_\ell(t) + R_\ell(t)$. We partition mobility and epidemic events and perform separate updates for each of them to sample the future state $\{S_\ell(t + \Delta t), E_\ell(t + \Delta t), I_\ell(t + \Delta t), R_\ell(t + \Delta t)\}_{\ell=1,\ldots,\mathcal{C}}$.

-*Mobility:*  The first step involves the calculation, for all sub-populations, of the number of agents who move within a time interval $\Delta t$. These quantities, denoted by $\{\Delta X_\ell\}_{\ell=1,\ldots,\mathcal{C}}$, for $X = S, E, I, R$, are extracted from binomial distributions $\Delta X_\ell \sim \mathbf{B}\left(X_\ell(t), p_\ell^{\text{out}}\right)$, with $p_\ell^{\text{out}} := 1 - e^{-\Delta t \sum_j m_{\ell,j}}$. Then, traveling agents have to be distributed among neighboring sub-populations. We call $\Delta X_{\ell,\ell'}$, respectively, the number of agents from compartment $X$ entering in sub-population $\ell'$ coming from $\ell$. Those numbers are sampled from the multinomial distributions, $\mathbf{M}(\Delta X_\ell; \{p_{\ell,\ell'}\}_{\ell'=1,\ldots,\mathcal{C}})$ with $p_{\ell,\ell'} := \dfrac{m_{\ell,\ell'}}{\sum_j m_{\ell,j}}$. The general multinomial distribution $\mathbf{M}(N; p_1, \ldots, p_k)$ is defined by the probabilities

$$P(n_1, \ldots, n_k) = \binom{N}{n_1 \cdots n_k} p_1^{n_1} \ldots p_k^{n_k}. \tag{S39}$$

One possible method for sampling numbers $\{n_1, \ldots, n_k\}$ from a multinomial distribution is by using an ordered sequence of binomial samples [75].

$$n_i \sim \mathbf{B}\left(N - \sum_{j<i} n_j, \frac{p_i}{1 - \sum_{j<i} p_j}\right), \quad i = 1, \ldots, k. \tag{S40}$$

At this point, the state of the system is updated with the mobility events:

$$X_\ell(t) \leftarrow X_\ell(t) + \sum_j \Delta X_{j,\ell}, \tag{S41}$$

but time is not yet increased, as the changes due to epidemic dynamics still need to be accounted for.

-*Epidemics:* Once agents have been reallocated according to the mobility dynamics [Eq. (S41)], occupation numbers are updated following the epidemic rules in [Eq. (45)]. To do so, we extract the binomial numbers:

$$\begin{aligned}
\Delta n_{\ell, S \to E} &\sim \mathbf{B}\left[S_\ell(t), 1 - \exp\left(-\beta \frac{I_\ell(t)}{N_\ell(t)} \Delta t\right)\right], \\
\Delta n_{\ell, E \to I} &\sim \mathbf{B}\left(E_\ell(t), 1 - e^{-\gamma \Delta t}\right), \\
\Delta n_{\ell, I \to R} &\sim \mathbf{B}\left(I_\ell(t), 1 - e^{-\mu \Delta t}\right),
\end{aligned} \tag{S42}$$

The new state of the system reads,

$$\begin{aligned}
S_\ell(t + \Delta t) &= S_\ell(t) - \Delta n_{\ell, S \to E}, \\
E_\ell(t + \Delta t) &= E_\ell(t) + \Delta n_{\ell, S \to E} - \Delta n_{\ell, E \to I}, \\
I_\ell(t + \Delta t) &= I_\ell(t) + \Delta n_{\ell, E \to I} - \Delta n_{\ell, I \to R}, \\
R_\ell(t + \Delta t) &= R_\ell(t) + \Delta n_{\ell, I \to R}.
\end{aligned} \tag{S43}$$

Finally, time is updated $t \to t + \Delta t$.