

RESEARCH

Open Access



# Tensor product algorithms for inference of contact network from epidemiological data

Sergey Dolgov<sup>1†</sup> and Dmitry Savostyanov<sup>2\*†</sup>

<sup>†</sup>Sergey Dolgov and Dmitry Savostyanov have contributed equally to this work.

\*Correspondence:  
D.Savostyanov@essex.ac.uk

<sup>1</sup> University of Bath, Claverton Down, Bath BA2 7AY, UK

<sup>2</sup> University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

## Abstract

We consider a problem of inferring contact network from nodal states observed during an epidemiological process. In a black-box Bayesian optimisation framework this problem reduces to a discrete likelihood optimisation over the set of possible networks. The cardinality of this set grows combinatorially with the number of network nodes, which makes this optimisation computationally challenging. For each network, its likelihood is the probability for the observed data to appear during the evolution of the epidemiological process on this network. This probability can be very small, particularly if the network is significantly different from the ground truth network, from which the observed data actually appear. A commonly used stochastic simulation algorithm struggles to recover rare events and hence to estimate small probabilities and likelihoods. In this paper we replace the stochastic simulation with solving the chemical master equation for the probabilities of all network states. Since this equation also suffers from the curse of dimensionality, we apply tensor train approximations to overcome it and enable fast and accurate computations. Numerical simulations demonstrate efficient black-box Bayesian inference of the network.

**Keywords:** Epidemiological modelling, Networks, Tensor train, Stochastic simulation algorithm, Markov chain Monte Carlo, Bayesian inference

**Mathematics Subject Classification:** 15A69, 34A30, 37N25, 60J28, 62F15, 65F55, 90B15, 95C42

## Introduction

### Background

The recent outbreak of COVID-19 and the public discussion that followed has led to better understanding of the central role that epidemiological models play in the decision making process and developing an informed response strategy. The quality of the mathematical models used in this process is crucial, not only because it allows to accurately predict the spread of a disease in population, but also in order to increase public trust in research and the decisions that are based on it. A large number of epidemic models used in education and research follow the Kermack–McKendrick compartmental SIR model [1]. An implicit assumption of this model is that the susceptible, infected and recovered groups are well-mixed in the sense that every person in any group has the



same probability to come in contact with anyone from another group, i.e. the contact network is homogeneous. However, this assumption holds relatively poorly in communities, which poses a significant limitation to application of compartmental models. More accurate models should include the information on the structure of contact network, leading to study of epidemics on networks [2–6].

Predicting the evolution of epidemic on a given network is much more difficult than solving a homogeneous model. Consider the case when each network node represents a single individual, who can be susceptible or infected. The dynamics of the epidemic becomes stochastic and depends on the position of infected individuals in the network and the number of susceptible neighbours they are in contact with. Hence, we need to model a probability distribution function of network states instead of states themselves. This probability distribution function satisfies the chemical master equation (CME) [7], which is an ordinary differential equation on the probability values. However, the total number of network states (and hence the size of the CME) grows exponentially in the number of network nodes [3, 4]. This makes the direct solution of the stochastic network models computationally intractable for large networks [6].

Perhaps the most traditional method for tackling the CME is the Stochastic Simulation Algorithm (SSA) [8] and variants, which compute random walks over the network states, distributed according to the CME solution. However, as a Monte Carlo method, the SSA is known to converge slowly, especially for rare events. Alternative approaches include mean-field approximations [9, 10], effective degree models [11–13], and edge-based compartmental models [14], but these models are approximate and rely on truncation of the state space. This introduces a truncation error that is difficult to estimate and/or keep below a desired tolerance for a general network. Other approaches include changing the original model into a surrogate model such as birth-death processes [15], or using neural networks [16, 17]. For solving the original model in a numerically controllable approximation framework, a new approach based on tensor product factorisations was recently proposed by authors in [18].

If the contact network is not known, we can attempt to solve an inverse problem, i.e. to infer the network from observations of disease data over time. For  $N$  network nodes, the number of possible networks grows exponentially in  $N^2$ . Hence, for large  $N$ , the problem complexity typically grows much faster than the information available, and network inference becomes a (very) under-determined problem [19]. Network inference is therefore only solved directly for very small population sizes [20, 21], or equivalently by assuming that the population consists of a few densely connected groups and estimating couplings between them [22]. Networks with mass-action kinetics can be inferred uniquely by observing transition rates at a simplex set of states [23], but the cardinality of this set is combinatorial in  $N$ . To address the problem for larger  $N$ , one can involve additional information about the network structure, such as degree distribution and sparsity [24, 25], community structure [26], and/or assumptions on the underlying statistical distribution for the network and infer its parameters [27]. In some cases more complex than pairwise interactions improve the network reconstruction [28]. Predictability of a stochastic process of observations and reconstructability of a network using their mutual information was considered in [29]. This information can be used to estimate the success of a network inference, before the full inference algorithm is applied.

In contrast to stochastic inference, one can instead model a deterministic dynamics and minimise the observational error over the network parameters in the right-hand side of the dynamics [30]. Another approach is to infer properties of the network rather than its exact structure [31], e.g. to find a class of network distributions, which the contact network most likely belongs to [15]. Related work include inferring the origin of epidemic given the contact network [32]. For a recent survey on network inference see [33]. Finally, most close to our work is the maximum-likelihood estimation of the network link probabilities from binary time series [34]. However, the latter paper uses an expectation-maximisation algorithm assuming the Poisson distribution, while in this paper we rely on the full Bayesian formalism with likelihoods computed directly from the chemical master equation.

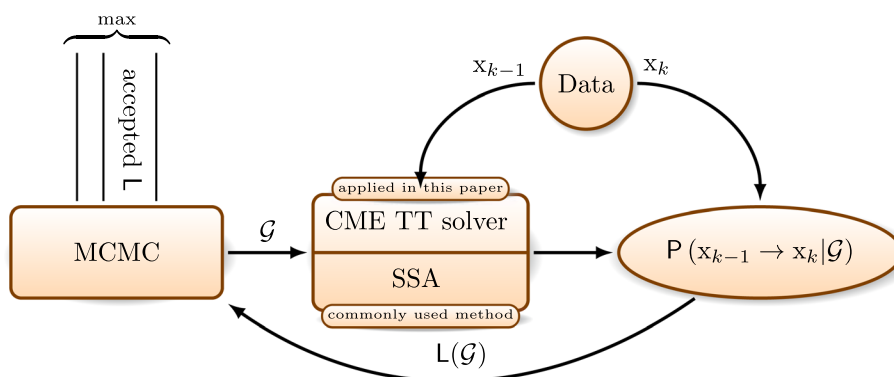
### Our contribution

In this paper we investigate inference of the contact network from states of the network observed over time. We use a Bayesian formulation to find the network with the maximum a posteriori (MAP) estimate. However, we do not assume any prior knowledge on the structure of the contact network (uniform prior), and thus look for the maximum likelihood estimate (MLE),

$$\text{network}_{\text{opt}} = \arg \max_{\text{network}} \mathbf{P}(\text{data} \mid \text{network}),$$

which we solve as black-box optimisation problem. To summarise the above, the network inference problem is difficult due to the following reasons.

1. Large inverse problem. The inverse problem is a high-dimensional discrete optimisation problem. Indeed, in an undirected network with  $N$  nodes there are  $\frac{1}{2}N(N-1)$  potential links, which are the optimisation variables, each of which can independently be in one of two states (on/off). In total, the search space which consists of  $2^{N(N-1)/2}$  possible networks, hence the exhaustive search is computationally unfeasible. Since the optimisation variables are discrete, the gradient of the target function is unavailable, and hence we can't apply steepest descent or Newton–Raphson algorithms. Hence, to find the near-optimal solution, we need to explore the structure of the high-dimensional array with some (possibly heuristic) algorithm, which may require a large number of target function evaluations.
2. Large forward problem. For each network in the search space, a single evaluation of the target function requires solving a forward problem, i.e. finding a probability of given data to be observed during the evolution of the disease on the current network. This is a Markov chain problem on the state space of accessible network states, which scales exponentially with the number of nodes, causing the currently available algorithms to struggle.
3. Low contrast caused by insufficient data. We may observe conditional probabilities to be (almost) the same,  $\mathbf{P}(\text{data} \mid \text{network}_1) \approx \mathbf{P}(\text{data} \mid \text{network}_2)$ , for example if the two networks differ only by the links attached to the nodes, for which we do not have (enough) events in the observed dataset. In this case, the Bayesian optimisation won't be able to choose one particular network, as a large number of them are equally likely to produce the observed dataset. Any numerical approximation errors due to limita-



**Fig. 1** Network inference workflow. An MCMC algorithm samples proposal network configurations,  $\mathcal{G}$ . The CME is solved on each time interval  $[t_{k-1}, t_k]$  in the observed data, starting from the state observation  $X(t_{k-1}) = x_{k-1}$  and obtaining the TT approximation of the probability of observing the state  $X(t_k) = x_k$  for the given network  $\mathcal{G}$ . The CME solver is applied in place of a more commonly used SSA method, that struggles to recover rare events. The probabilities for all data are multiplied to form the likelihood  $L(\mathcal{G})$ , which is accepted or rejected in the MCMC. Finally, the network with the maximum likelihood among the MCMC samples is inferred

tions of the forward problem solvers (see item 2) add extra noise to the high-dimensional probability density function and further complicate the optimisation process (see item 1). Due to the probabilistic nature of the problem, the ground truth network  $\text{network}_*$ , for which the observed dataset was generated, may differ from the optimal network recovered by the Bayesian inference method,  $\text{network}_* \neq \text{network}_{\text{opt}}$ .

4. High contrast caused by a large amount of data. Although adding more data makes the ground truth network a unique global optimum for Bayesian optimisation, it also creates a large number of local optima. In a black-box optimisation setting, there is no prior information that could navigate the optimisation towards the global optimum, and the algorithm can be trapped in a local optimum for considerable time.

The existing literature often does not consider these issues separately, nor approach the problem directly. It is typically stated that the network optimisation is impossible to solve, and alternative formulations are considered [6, 15, 19, 33]. In this paper we attempt to perform the black-box network inference directly following the Bayesian optimisation framework. To tackle the forward problem, we solve the CME using tensor product factorisations [18]. A related work was recently proposed in tensor network community, but it is limited to linear one-dimensional chains [35]. We demonstrate that the proposed method provides faster and more accurate solution to the forward problem compared to SSA, particularly when the network is far from optimal.

Next, we apply two Markov Chain Monte Carlo (MCMC) algorithms for black-box discrete high-dimensional optimisation and analyse results. An overall workflow of this procedure is illustrated in Fig. 1. By simulating three examples of networks, we show that by collecting sufficiently many data we can make the contrast high enough to infer the original ground truth network.

**Notation**

We use calligraphic font for contact networks  $\mathcal{G}$ , blackboard font for sets  $\mathbb{R}, \mathbb{G}, \mathbb{X}$ , sans-serif font for probability  $\mathbb{P}$ , expectation  $\mathbb{E}$ , variance  $\mathbb{V}$ , and likelihood  $\mathbb{L}$ . Indices  $m, n$  and scalar values  $x, p$  are shown in usual maths italics. We use maths roman font for vectors, e.g. network states  $x$  or unit vectors  $e_n$ . We use capitals for matrices, e.g. adjacency matrix  $G$  of network  $\mathcal{G}$ . When the considered vectors and matrices grow exponentially in number of people  $N$ , and hence suffer from the curse of dimensionality, we highlight it using bold font, e.g.  $\mathbf{p}$  for high-dimensional probability distribution function, and  $\mathbf{A}$  for the matrix of CME.

**Methods**

**Forward problem:  $\varepsilon$ -SIS epidemic on network**

We consider the  $\varepsilon$ -SIS model of the contact process, which is a variation of a classical susceptible-infected-susceptible (SIS) model, allowing for every node to self-infect with rate  $\varepsilon$ . This process was originally proposed by Hill et al. to describe the spread of emotions in social network [36]. Mieghem et al. studied analytical properties of the model for fully connected networks [37, 38]. Zhang et al. extended this study to arbitrary networks and found conditions under which the equilibrium distribution can be accurately approximated by a scaled SIS process, gaining useful insights on vulnerability of the population [39].

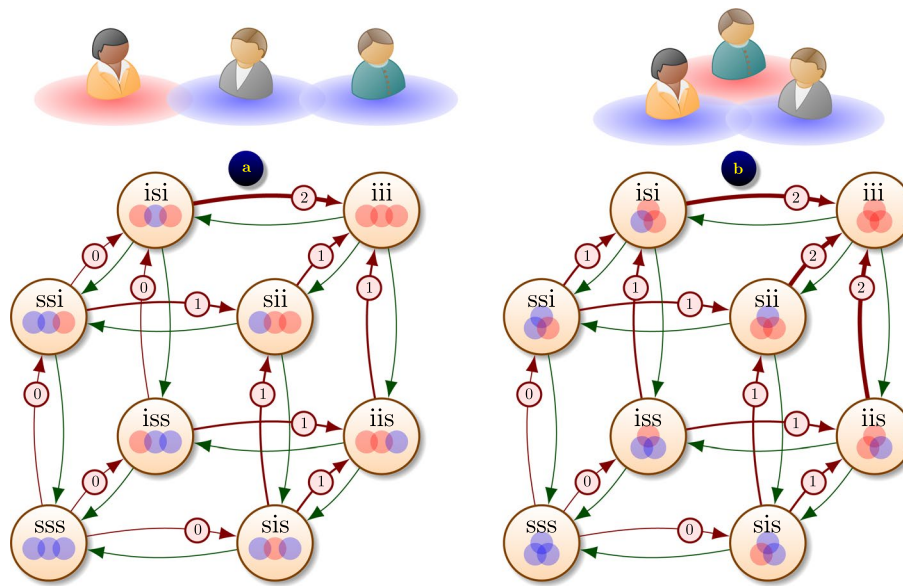
The classical SIS model has an absorbing state where all nodes are susceptible (i.e. the network is fully healthy), but due to spontaneous self-infections the  $\varepsilon$ -SIS model does not have an absorbing state, hence the epidemics lasts forever. This property allows us to observe the epidemics for sufficiently long time and eventually collect the dataset which is large enough to ensure the required contrast for the Bayesian optimisation.

We consider a  $\varepsilon$ -SIS epidemic on a unweighted simple network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , which is a set of nodes (representing people, or agents)  $\mathcal{V} = \{1, 2, \dots, N\}$  and links (or edges, representing contacts between agents)  $\mathcal{E} = \{(m, n) : m \in \mathcal{V}, n \in \mathcal{V}, m \neq n\}$ . We additionally assume that the contacts are bidirectional, i.e.  $(m, n) \in \mathcal{E} \Leftrightarrow (n, m) \in \mathcal{E}$ , which allows us to introduce a symmetric adjacency relation  $m \sim n \Leftrightarrow (m, n) \in \mathcal{E}$  for the connections.

Each node can be in one of two states,  $x_n \in \mathbb{X} = \{\text{susceptible}, \text{infected}\} = \{0, 1\}$ , for  $n \in \mathcal{V}$ . The state of the whole network is therefore a vector  $x = (x_1 \ x_2 \ \dots \ x_N)^T \in \mathbb{X}^N$ . We consider the system dynamics as a continuous-time Markov jump process on the state space  $\Omega = \mathbb{X}^N$ . The following two types of transitions (or reactions), infection and recovery, occur independently and at random according to the Poisson process with the following rates

$$p_{x \rightarrow y} = \begin{cases} p_{x \rightarrow y}^{(\text{inf})}, & \text{if } \exists n \in \mathcal{V} : y = x + e_n; \\ p_{x \rightarrow y}^{(\text{rec})}, & \text{if } \exists n \in \mathcal{V} : y = x - e_n; \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where  $e_n \in \mathbb{R}^N$  is the  $n$ -th unit vector. For simplicity, we assume that the recovery rates  $p_{x \rightarrow y}^{(\text{rec})} = \gamma$  are the same for all nodes of the network. In the classical SIS model, the infection rate for the susceptible node  $x_n = 0$  is proportional to the number of its infected neighbours,  $I_n(x) = |\{m \in \mathcal{V} : m \sim n, x_m = 1\}|$ , and the per-contact rate  $\beta$ , which we also consider the same across all network. In the  $\varepsilon$ -SIS model, an additional infection rate  $\varepsilon$  is introduced to describe possible infection through contacts with the external, off-the-network, environment. Hence, the infection rate is  $p_{x \rightarrow y}^{(\text{inf})} = I_n(x)\beta + \varepsilon$ . Examples of the Markov transition graph are shown in Fig. 2.



**Fig. 2** Markov transitions between network states: **a**  $\varepsilon$ -SIS epidemic on a chain of  $N = 3$  people; **b**  $\varepsilon$ -SIS epidemic in a fully connected network of  $N = 3$  people. On the graph, green arrows denote recovery process with rate  $\gamma$ , and red arrows with a circled number  $k$  denote infection process with rate  $k\beta + \varepsilon$

The stochastic properties of the system are described as probabilities of network states  $p(x, t) = P(\text{system is in state } x \text{ at time } t)$ . The system dynamics is written as a system of ordinary differential equations (ODEs), known as *Markovian master equation* [3, 7], Chapman or *forward Kolmogorov equations*:

$$p'(x, t) = \sum_{y \in \mathbb{X}^N} (p_{y \rightarrow x} \cdot p(y, t) - p_{x \rightarrow y} \cdot p(x, t)), \quad x \in \mathbb{X}^N, \tag{2}$$

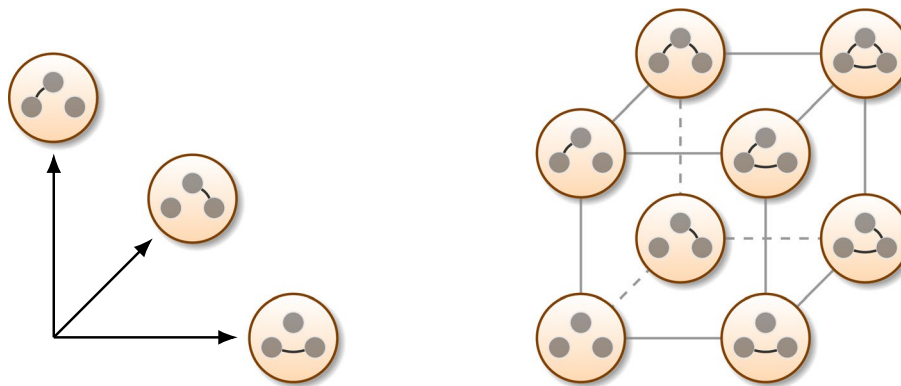
subject to initial conditions  $p(x_0, 0) = 1$  for the initial state  $x = x_0$  and  $p(x, 0) = 0$  otherwise. The number of ODEs scales as  $2^N$ , making traditional numerical solvers struggle for even moderate values of  $N$ .

**Inverse problem: Bayesian inference of the network**

Our goal is to infer the most probable contact network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  from the observed data  $\mathcal{X} = \{t_k, x(t_k)\}_{k=0}^K$ . According to the Bayes theorem [40],  $P(\mathcal{G}|\mathcal{X}) = \frac{P(\mathcal{X}|\mathcal{G})P(\mathcal{G})}{P(\mathcal{X})}$ , where  $P(\mathcal{G})$  is the *prior* probability distribution for the grid,  $P(\mathcal{X}|\mathcal{G})$  is the *likelihood* of the observed data as a function of the grid  $\mathcal{G}$ ,  $P(\mathcal{X})$  is the probability of the observed data, and  $P(\mathcal{G}|\mathcal{X})$  is the posterior probability of the grid given the data.

The connectivity network  $\mathcal{G}$  can be described by its adjacency matrix  $G = [g_{m,n}]_{m,n \in \mathcal{V}}$  with  $g_{m,n} = 1 \Leftrightarrow (m, n) \in \mathcal{E}$  and  $g_{m,n} = 0$ , otherwise. Since  $\mathcal{G}$  is simple,  $G$  is a binary symmetric matrix,  $G = G^T$ , with zero diagonal,  $\text{diag}(G) = 0$ . It is easy to see that a grid  $\mathcal{G}$  can be described by  $\frac{1}{2}N(N - 1)$  independent binary variables  $g_{m,n} \in \mathbb{B} = \{0, 1\}$  with  $m, n \in \mathcal{V}$  and  $m > n$ , representing the states (on/off) of possible edges  $(m, n) \in \mathcal{E}$ . Therefore, for a fixed number of nodes  $|\mathcal{V}| = N$ , the set of all possible networks  $\mathbb{G} = \{\mathcal{G} : g_{m,n} = g_{n,m} \in \mathbb{B}, m, n \in \mathcal{V}, m > n\}$  has cardinality  $|\mathbb{G}| = 2^{N(N-1)/2}$ . Note that





**Fig. 3** The set of all possible networks with  $N = 3$  nodes is a binary hypercube in dimension  $\frac{1}{2}N(N - 1) = 3$

$\mathbb{G}$  is isomorphic to  $\mathbb{B}^{N(N-1)/2}$ , the set of adjacency elements  $g_{m,n}$ . The structure of this set is illustrated in Fig. 3.

Assuming that nodes are known and no prior information of the edges  $\mathcal{E}$  is available, we take the uniform prior distribution,  $P(\mathcal{G}) = 2^{-N(N-1)/2}$ . Although  $P(\mathcal{X})$  is unknown, it does not affect the optimisation problem, as  $\max_{\mathcal{G}} P(\mathcal{G}|\mathcal{X}) = \frac{2^{-N(N-1)/2}}{P(\mathcal{X})} \max_{\mathcal{G}} P(\mathcal{X}|\mathcal{G})$ , hence  $\arg \max_{\mathcal{G}} P(\mathcal{G}|\mathcal{X}) = \arg \max_{\mathcal{G}} P(\mathcal{X}|\mathcal{G})$ .

Due to the Markovian property of the system dynamics, the likelihood can be expanded as follows,

$$\begin{aligned} L(\mathcal{G}) &= P(\mathcal{X}|\mathcal{G}) = P(X(t_1) = x_1, \dots, X(t_K) = x_K|\mathcal{G}) \\ &= P(X(t_1) = x_1|X(t_0) = x_0, \mathcal{G}) \cdots P(X(t_K) = x_K|X(t_{K-1}) = x_{K-1}, \mathcal{G}), \\ &= \prod_{k=1}^K \underbrace{P(X(t_k) = x_k|X(t_{k-1}) = x_{k-1}, \mathcal{G})}_{P(x_{k-1} \rightarrow x_k|\mathcal{G})}, \end{aligned} \tag{3}$$

where  $X(t) \in \mathbb{X}^N$  are random variables describing the network states during its stochastic evolution, and the time sequence  $\{t_k\}_{k=0}^K$  is monotonically increasing. We see that the likelihood is a product of transition probabilities  $P(x_{k-1} \rightarrow x_k|\mathcal{G}) = P(X(t_k) = x_k|X(t_{k-1}) = x_{k-1}, \mathcal{G})$ , which are the probabilities for the system to evolve from the state  $x_{k-1}$  to  $x_k$  over the time period  $[t_{k-1}, t_k]$ . The (black-box) Bayesian network inference therefore boils down to likelihood optimisation,

$$\mathcal{G}_{\text{opt}} = \arg \max_{\mathcal{G} \in \mathbb{G}} \log L(\mathcal{G}) = \arg \max_{\mathcal{G} \in \mathbb{G}} \sum_{k=1}^K \log P(x_{k-1} \rightarrow x_k|\mathcal{G}). \tag{4}$$

To compute a single log-likelihood  $\log L(\mathcal{G})$  in (4), we need to solve  $K$  forward problems, i.e. estimate the chance of arriving to the state  $x_k$  from the state  $x_{k-1}$  over the period of time  $t \in [t_{k-1}, t_k]$  for  $k = 1, \dots, K$ . To find the optimal network  $\mathcal{G}_{\text{opt}}$ , we may need to compute a large amount of log-likelihoods for different networks  $\mathcal{G}$ , hence the efficiency of the forward solver is crucial to make the optimisation procedure feasible. This rules out a possibility of solving (2) directly due to the curse of dimensionality.

### Stochastic simulation algorithms for forward problem

Traditionally, probabilities  $p(x, t)$  are estimated using the Stochastic Simulation Algorithm (SSA) [8], or some more efficient (e.g. multilevel) Monte Carlo simulations of the realisations of the model [41–44]. Essentially, these methods sample  $N_{\text{SSA}}$  random walks through the state space  $\Omega = \mathbb{X}^N$  starting at the initial state  $x_{k-1}$ , count the number  $n_{\text{SSA}}$  of trajectories that end up in the state  $x_k$  by the time  $t_k$ , and estimate the target probability as a frequency,

$$P(x_{k-1} \rightarrow x_k | \mathcal{G}) \approx \tilde{P}(x_{k-1} \rightarrow x_k | \mathcal{G}) = \frac{n_{\text{SSA}}}{N_{\text{SSA}}}. \quad (5)$$

The cost of sampling each trajectory does not grow exponentially with  $N$  which makes these methods free from the curse of dimensionality. However, the convergence of these algorithms is not particularly fast, with typical estimates

$$\text{err} = \left| P(x_{k-1} \rightarrow x_k | \mathcal{G}) - \tilde{P}(x_{k-1} \rightarrow x_k | \mathcal{G}) \right| \leq c N_{\text{SSA}}^{-\delta},$$

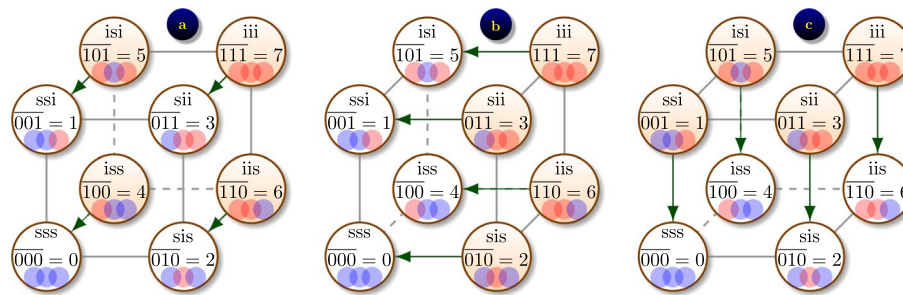
with  $0.5 \leq \delta \leq 1$  depending on a particular method. This is usually sufficient to estimate large probabilities and main statistics (such as mean and variance) of the process. However, if the probability  $p = P(x_{k-1} \rightarrow x_k | \mathcal{G})$  is small, to ensure the desired relative precision  $|p - \tilde{p}| \leq \epsilon p$ , the number of samples should satisfy  $c N_{\text{SSA}}^{-\delta} \leq \epsilon p$ , or  $N_{\text{SSA}} \geq (\epsilon p / c)^{-1/\delta}$ . In practice this means that estimation of rare events with probabilities  $p \lesssim 10^{-6}$  with these algorithms can be prohibitively expensive.

If the computational budget of  $N_{\text{SSA}}$  trajectories is exhausted and none of them arrived at  $x_k$ , then  $n_{\text{SSA}} = 0$  and  $\tilde{p} = 0$ , i.e. the event is not resolved with the algorithm and is considered impossible. If this happens for at least one transition from state  $x_{k-1}$  to  $x_k$  for some  $k = 1, \dots, K$ , then the whole likelihood  $L(\mathcal{G}) = 0$  for the given network  $\mathcal{G}$ . In practical computations this issue can occur for most networks  $\mathcal{G} \neq \mathcal{G}_*$  except the ‘ground truth’ network and its close neighbours. The limited computation budget for the forward problem therefore leads to flattening of the high-dimensional landscape of the likelihoods, wiping off the structural information that should be used to navigate the optimisation algorithm towards the solution of (4). This motivates the development of more accurate methods for the forward problem, such as the tensor product approach that we discuss in the following subsection.

### Tensor product algorithms for forward problem

To find the probabilities composing the likelihood (4), we can solve the system of ODEs (2) for the probabilities of the network states. This system is commonly known as the chemical master equation (CME) and consists of  $|\mathbb{X}^N| = 2^N$  equations and unknowns, hence traditional solvers suffer from the curse of dimensionality. To mitigate this problem, different approaches were used, including sparse grids [45], adaptive finite state projections [46–48], radial basis functions [49], neural networks [16, 17], and tensor product approximations, such as canonical polyadic (CP) format [50–52], and more recently tensor train (TT) format [18, 53–60]. Here we briefly describe the tensor train approach used in our recent paper [18].





**Fig. 4** The tensor product structure of recovery transitions  $[p_{x \rightarrow (x-e_n)}^{(rec)}]_{x \in \mathbb{X}^N}$  is illustrated for population of  $N = 3$  people. Recovery takes place on individual nodes and hence does not depend on contact network. In each panel, highlighted states  $x$  are where  $p_{x \rightarrow (x-e_n)}^{(rec)} = \gamma$ , indicating that person  $n$  is infected and can recover; this is also shown by green arrows. Non-highlighted states correspond to  $p_{x \rightarrow (x-e_n)}^{(rec)} = 0$ . **a**  $n = 1$ , **b**  $n = 2$ , **c**  $n = 3$

First, we note that among  $2^N$  reaction rates  $p_{y \rightarrow x}$  in (2) only  $2N$  are non-zero, according to (1):

$$p'(x, t) = \sum_{n=1}^N \left( p_{(x-e_n) \rightarrow x}^{(inf)} p(x - e_n, t) + p_{(x+e_n) \rightarrow x}^{(rec)} p(x + e_n, t) - \left( p_{x \rightarrow (x+e_n)}^{(inf)} + p_{x \rightarrow (x-e_n)}^{(rec)} \right) p(x, t) \right), \tag{6}$$

for  $x \in \mathbb{X}^N$ . Let us now uncover the tensor product structure of the matrix of this CME. For this we assume that  $2^N$  network states  $x \in \mathbb{X}^N$  are ordered lexicographically, i.e. the state  $x = (x_1, \dots, x_N)^T$  has index

$$\bar{x} = \overline{x_1 x_2 \dots x_N} = 2^{N-1} x_1 + 2^{N-2} x_2 + \dots + 2^0 x_N,$$

which corresponds to how binary numbers are written in big-endian notation, e.g.  $\overline{000} = 0, \overline{001} = 1, \overline{010} = 2, \overline{100} = 4, \overline{101} = 5, \overline{111} = 7$ , see also Fig. 4. When the probability distribution function (p.d.f.) vector  $\mathbf{p}(t) = [p(x, t)]_{x \in \mathbb{X}^N}$  is composed, we place the probability  $p(x, t)$  in position  $\bar{x}$ , assuming 0-based indexing scheme, i.e. vector indices enumerated from 0 up to  $2^N - 1$ . If 1-based indexing scheme is used, we place  $p(x, t)$  in position  $\bar{x} + 1$ .

Using *indicator* function

$$\mathbf{1}_{\text{condition}} = \begin{cases} 1, & \text{if condition is true} \\ 0, & \text{if condition is false,} \end{cases}$$

we can write  $p_{x \rightarrow (x-e_n)}^{(rec)} = \gamma \mathbf{1}_{x_n=1}$ , and  $p_{x \rightarrow (x+e_n)}^{(inf)} = (\varepsilon + \beta \sum_{m \sim n} \mathbf{1}_{x_m=1}) \mathbf{1}_{x_n=0}$ . Collecting these reaction rates in vectors of size  $2^N$ , and using the big-endian lexicographic ordering as explained above, we obtain tensor product decomposition

$$[p_{x \rightarrow (x-e_n)}^{(rec)}]_{x \in \mathbb{X}^N} = \gamma \vec{e} \otimes \dots \otimes \vec{e} \otimes \vec{i} \otimes \vec{e} \otimes \dots \otimes \vec{e}, \tag{7}$$

where  $\vec{i} = (0 \ 1)^T$  appears in position  $n$ ,  $\vec{e} = (1 \ 1)^T$  appear elsewhere. The tensor product structure is illustrated for  $N = 3$  in Fig. 4. For example, the full vector of recovery transitions corresponding to recovery of person  $n = 1$  is

$$\begin{aligned}
 \left[ p_{x \rightarrow (x-e_1)}^{(\text{rec})} \right]_{x \in \mathbb{X}^N} &= \gamma \vec{i} \otimes \vec{e} \otimes \vec{e} \\
 &= \gamma (0 \ 1)^T \otimes (1 \ 1)^T \otimes (1 \ 1)^T \\
 &= \gamma (0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1)^T.
 \end{aligned} \tag{8}$$

We can see that this transition is possible from states  $x = (x_1 \ x_2 \ x_3)$  with  $x_1 = 1$ , which are located in positions  $\overline{100} = 4$ ,  $\overline{101} = 5$ ,  $\overline{110} = 6$ ,  $\overline{111} = 7$  of the vector (assuming 0-based indexing), in agreement to Eq. (8) and Fig. 4a.

Similarly,

$$\begin{aligned}
 \left[ p_{x \rightarrow (x+e_n)}^{(\text{inf})} \right]_{x \in \mathbb{X}^N} &= \varepsilon \vec{e} \otimes \dots \otimes \vec{e} \otimes \vec{s} \otimes \vec{e} \otimes \dots \otimes \vec{e} \\
 &\quad + \beta \sum_{m \sim n} \vec{e} \otimes \dots \otimes \vec{e} \otimes \vec{s} \otimes \vec{e} \otimes \dots \otimes \vec{e} \otimes \vec{i} \otimes \vec{e} \otimes \dots \otimes \vec{e},
 \end{aligned} \tag{9}$$

where  $\vec{s} = (1 \ 0)^T$  appears in position  $n$ ,  $\vec{i} = (0 \ 1)^T$  appear in positions  $m \sim n$ ,  $\vec{e} = (1 \ 1)^T$  appear elsewhere. To complete the expansion for the right-hand side of (6), we need to express the shifted state  $p(x - e_n, t)$  as a sum over probabilities  $p(y, t)$  as follows

$$\begin{aligned}
 p(x - e_n, t) &= \sum_{y \in \mathbb{X}^N} 1_{x_1=y_1} \dots 1_{x_{n-1}=y_{n-1}} \cdot 1_{x_n-1=y_n} \cdot 1_{x_{n+1}=y_{n+1}} \dots 1_{x_N=y_N} \cdot p(y, t), \\
 \left[ p(x - e_n, t) \right]_{x \in \mathbb{X}^N} &= \underbrace{\left( \text{Id} \otimes \dots \otimes \text{Id} \otimes J^T \otimes \text{Id} \otimes \dots \otimes \text{Id} \right)}_{J_n^T} \mathbf{p}(t),
 \end{aligned} \tag{10}$$

where the *shift matrix*  $J^T = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$  appears in position  $n$ , and identity matrices  $\text{Id} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  appear elsewhere. Similarly,  $\left[ p(x + e_n, t) \right]_{x \in \mathbb{X}^N} = \mathbf{J}_n \mathbf{p}$  with  $\mathbf{J}_n = \text{Id} \otimes \dots \otimes \text{Id} \otimes J \otimes \text{Id} \otimes \dots \otimes \text{Id}$ . Combining the above, we collect all equations of (6) in a vectorised CME

$$\mathbf{p}'(t) = \mathbf{A} \mathbf{p}(t), \quad \mathbf{p}(0) = \mathbf{p}_0, \tag{11}$$

where the  $2^N \times 2^N$  matrix  $\mathbf{A}$  admits the following tensor product form:

$$\begin{aligned}
 \mathbf{A} &= \gamma \sum_{n=1}^N \text{Id} \otimes \dots \otimes \text{Id} \otimes (J - \text{Id}) \hat{I} \otimes \text{Id} \otimes \dots \otimes \text{Id} \\
 &\quad + \varepsilon \sum_{n=1}^N \text{Id} \otimes \dots \otimes \text{Id} \otimes (J^T - \text{Id}) \hat{S} \otimes \text{Id} \otimes \dots \otimes \text{Id} \\
 &\quad + \beta \sum_{n=1}^N \sum_{m \sim n} \text{Id} \otimes \dots \otimes \text{Id} \otimes (J^T - \text{Id}) \hat{S} \otimes \text{Id} \otimes \dots \otimes \text{Id} \otimes \hat{I} \otimes \text{Id} \otimes \dots \otimes \text{Id},
 \end{aligned} \tag{12}$$

where  $\hat{I} = \text{diag}(\vec{i}) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$  and  $\hat{S} = \text{diag}(\vec{s}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ . This so-called canonical polyadic (CP) [61–63] form represents the CME matrix  $\mathbf{A}$  as a sum of  $(2N + |\mathcal{E}|)$  elementary tensors, each of which is a direct product of  $N$  small  $2 \times 2$  matrices, acting on a single node

of the network only. Hence, the storage for  $A$  reduces from  $\mathcal{O}(2^N \langle k \rangle)$  down to  $\mathcal{O}((2N + \langle k \rangle)N)$  elements, where  $\langle k \rangle = |\mathcal{E}|/|\mathcal{V}|$  denotes the average degree of  $\mathcal{G}$ . The curse of dimensionality for the matrix is therefore removed.

To remove the exponential complexity in solving (11), we need to achieve a similar compression for the p.d.f.  $\mathbf{p}(t) = [p(\mathbf{x}, t)]_{\mathbf{x} \in \mathbb{X}^N}$ , for which we employ the *tensor train* (TT) format [64].

$$\mathbf{p} \approx \tilde{\mathbf{p}} = \sum_{\alpha_0, \dots, \alpha_N=1}^{r_0, \dots, r_N} \mathbf{p}_{\alpha_0, \alpha_1}^{(1)} \otimes \dots \otimes \mathbf{p}_{\alpha_{n-1}, \alpha_n}^{(n)} \otimes \dots \otimes \mathbf{p}_{\alpha_{N-1}, \alpha_N}^{(N)}. \tag{13}$$

Here, the  $r_{n-1} \times 2 \times r_n$  factors  $\mathbf{p}^{(n)} = [\mathbf{p}_{\alpha_{n-1}, \alpha_n}^{(n)}(\mathbf{x}_n)]$ ,  $n = 1, \dots, N$ , are called *TT cores*, and the ranges of the summation indices  $r_0, \dots, r_N$  are called *TT ranks*. Each core  $\mathbf{p}^{(n)}$  contains information related to node  $n$  in the network, and the summation indices  $\alpha_{n-1}, \alpha_n$  of core  $\mathbf{p}^{(n)}$  link it to cores  $\mathbf{p}^{(n-1)}$  and  $\mathbf{p}^{(n+1)}$ . The matrix-vector multiplication can be performed fully in tensor product format. Using recently proposed algorithms [55, 56] the linear system of ODEs (11) can be solved fully in the TT format avoiding the curse of dimensionality, as explained in details in [18].

**Ordering of network nodes for faster forward problem solving**

The TT decomposition of probability functions exhibits low ranks when distant (with respect to their position in the state vector) variables are weakly correlated; see e.g. [65] for a rigorous analysis of this for the multivariate normal probability density function. Numerical approaches to order the variables in such a way include a greedy complexity optimisation over a reduced space of permutations [66, 67], using gradients to compute a Fisher-type information matrix and its eigenvalue decomposition to sort or rotate the variables [68], and sorting the variables according to the Fiedler vector of the network [69]. Since in our case the variables are discrete, we adopt the latter approach.

We consider the Laplacian matrix of the network  $\mathcal{G}$ , defined as follows

$$L = \text{diag}(Ge) - G, \tag{14}$$

where  $G \in \mathbb{B}^{N \times N}$  is the adjacency matrix of  $\mathcal{G}$ , and  $\mathbf{e} = (1 \ 1 \ \dots \ 1)^T \in \mathbb{B}^N$ . We are particularly interested in the *Laplacian spectrum* of  $\mathcal{G}$ , i.e. solutions to the eigenvalue problem  $Lu = \lambda u$ . It is easy to see that since  $G = G^T$  we also have  $L = L^T$ , hence the spectrum is real,  $\lambda \in \mathbb{R}$ . We also note that  $L = [\ell_{m,n}]_{m,n \in \mathcal{V}}$  is diagonally dominant, since  $\ell_{m,m} = \sum_{n \in \mathcal{V}} g_{m,n} \geq 0$ , and  $\ell_{m,n} = -g_{m,n} \leq 0$  for  $m \neq n$ , hence  $|\ell_{m,m}| = \sum_{m \neq n} |\ell_{m,n}|$ . Since  $\text{diag}(L) \geq 0$  and  $L$  is symmetric and diagonally dominant, the Laplacian is positive semi-definite, hence its eigenvalues are nonnegative,  $\lambda_{n-1} \geq \lambda_{n-2} \geq \dots \geq \lambda_1 \geq \lambda_0 \geq 0$ . It is easy to see that  $Le = 0$ , hence  $\lambda_0 = 0$  is the lowest eigenvalue with the corresponding eigenvector  $\mathbf{u}_0 = \mathbf{e}$ .

The second eigenvalue, which we denote  $\lambda_1$ , is called the *algebraic connectivity* of  $\mathcal{G}$  or *Fiedler value*. It was known since [70] that  $\lambda_1 = 0$  if and only if  $\mathcal{G}$  is disconnected. This is a particularly easy scenario for epidemiological modelling. If  $\mathcal{G}$  consists of two disjoint networks,  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , then nodes from  $\mathcal{G}_1$  and  $\mathcal{G}_2$  can not affect each other. The random variables associated with those nodes are therefore independent,

i.e. if  $X_1 \in \mathcal{G}_1$  and  $X_2 \in \mathcal{G}_2$  then  $P(X_1, X_2) = P(X_1)P(X_2)$ . This means that if we order the variables  $x_1, \dots, x_N$  in such a way that the first block  $x_1, \dots, x_m \in \mathcal{G}_1$  and the second block  $x_{m+1}, \dots, x_N \in \mathcal{G}_2$  do not overlap, then the TT format (13) for the p.d.f.  $p(x_1, \dots, x_N)$  will have the TT rank  $r_m = 1$  for the connection separating  $\mathcal{G}_1$  and  $\mathcal{G}_2$ .

These geometric properties of the network can be estimated from the eigenvector  $v = u_1$ , also known as *Fiedler vector*. In the pioneering paper [71] it was related to finding an optimal cut in the network  $\mathcal{G}$ . It was generalised to reducing the envelope (or the bandwidth) of the adjacency matrix  $G$  in [72], and later to finding orderings of variables for which TT decomposition (13), and related MPS and DMRG representations in quantum physics [73], have lower TT ranks [69]. The Fiedler vector can be computed by minimising the *Rayleigh quotient*  $v = \arg \min_{v \perp e} v^T L v / \|v\|^2$ , also known as the *Courant minimax* principle, orthogonally to  $u_0 = e$ . Following [72], we use the Fiedler vector to define a one-dimensional embedding of the graph to a linear chain. Let  $\sigma \in \mathfrak{S}_n$  be the permutation vector of the set of nodes  $\mathcal{V}$  such that  $(v_{\sigma_i})$  is ordered, i.e.  $v_{\sigma_1} \geq v_{\sigma_2} \geq \dots \geq v_{\sigma_N}$ , or equivalently in the ascending order. Following [69], the same permutation of variables also reduces the TT ranks of the TT decomposition (13). In particular, it groups together variables corresponding to independent subnetworks. Hence, we compute this permutation and adopt its order of variables before solving the forward problem (11) using tensor product algorithms.

### Algorithms for Bayesian inverse problem

Since the full grid search is unfeasible for even moderate networks, we adopt the Metropolis-Hastings Markov Chain Monte Carlo (MCMC) method to approach the maximum of  $L(\mathcal{G})$ . The method is depicted in Algorithm 1. We need to choose a *proposal distribution*  $q(\hat{\mathcal{G}}|\mathcal{G})$  which is tractable for sampling a new state of the network  $\hat{\mathcal{G}}$ , given the current network  $\mathcal{G}$ . In each iteration, given the current network  $\mathcal{G}_i$ , we sample a new proposal  $\hat{\mathcal{G}}$  and accept or reject it with probability based on the Metropolis-Hastings ratio, forming a Markov Chain of network configurations  $\mathcal{G}_0, \mathcal{G}_1, \dots$ . After the Markov Chain is computed, we return the sample of this chain with the maximal likelihood.

**Algorithm 1** MCMC algorithm for the likelihood maximisation

- 
- 1: Choose the proposal distribution  $q(\hat{\mathcal{G}}|\mathcal{G})$ , initial network  $\mathcal{G}_0$  and length of the chain  $N_{\text{eval}}$ .
  - 2: **for**  $i = 0, \dots, N_{\text{eval}} - 1$  **do**
  - 3:     Sample a new proposal network  $\hat{\mathcal{G}} \sim q(\hat{\mathcal{G}}|\mathcal{G}_i)$ .
  - 4:     Compute the Metropolis-Hastings ratio  $h(\hat{\mathcal{G}}, \mathcal{G}_i) = \frac{L(\hat{\mathcal{G}}) q(\mathcal{G}_i|\hat{\mathcal{G}})}{L(\mathcal{G}_i) q(\hat{\mathcal{G}}|\mathcal{G}_i)}$ .  $\triangleright$  As in (15).
  - 5:     Sample a uniformly distributed random number  $r \sim \mathcal{U}(0, 1)$ .
  - 6:     **if**  $r < \min(h(\hat{\mathcal{G}}, \mathcal{G}_i), 1)$  **then**
  - 7:         Accept the proposal by setting  $\mathcal{G}_{i+1} = \hat{\mathcal{G}}$ .
  - 8:     **else**
  - 9:         Reject the proposal by setting  $\mathcal{G}_{i+1} = \mathcal{G}_i$ .
  - 10:    **end if**
  - 11: **end for**
  - 12: **return**  $\mathcal{G}_\star = \arg \max_{i=0, \dots, N_{\text{eval}}} \log L(\mathcal{G}_i)$ .
-

This algorithm is known to converge to the true distribution  $\frac{1}{Z}L(\mathcal{G})$  (where  $Z$  is the normalising constant) under mild assumptions [74]. We implement two proposal distributions.

1. Choose one link in  $\mathcal{G}_i$  uniformly at random and toggle its state (on/off). Since there are  $N(N-1)/2$  possible links to toggle,  $q(\hat{\mathcal{G}}|\mathcal{G}) = \frac{2}{N(N-1)}$  independently of  $\hat{\mathcal{G}}$  and  $\mathcal{G}$ , and hence cancels in the Metropolis-Hastings ratio. We will call Algorithm 1 with this proposal “MCMC-R”, since it samples the links with Replacement.
2. Every  $N(N-1)/2$  iterations (i.e. when  $\text{mod}(n, N(N-1)/2) = 0$ ), sample a random permutation vector  $\sigma \in \mathfrak{S}_{N(N-1)/2}$  of the set  $\{1, \dots, N(N-1)/2\}$ , and in the next  $N(N-1)/2$  iterations toggle links in the order prescribed by  $\sigma$ . This is still a valid MCMC algorithm with a constant proposal distribution, but now with respect to  $\sigma$ , corresponding to a collection of networks in consecutive update steps,  $(\mathcal{G}_{i+1}, \dots, \mathcal{G}_{i+N(N-1)/2})$ . In our numerical experiments reported in Sect. “Results” this algorithm sometimes increased the likelihood faster in terms of the individual link changes (and hence computing time), and resulted in a better grid reconstruction. In each block of  $N(N-1)/2$  iterations, this algorithm proposes link changes without replacement [75]. For this reason, we will call this algorithm “MCMC-noR” (for “no Replacement”).

Cancelling the constant proposal probability, and using log-likelihoods to avoid numerical over- and under-flow errors, we can rewrite the Metropolis-Hastings ratio as  $h(\hat{\mathcal{G}}, \mathcal{G}_i) = e^{\ln L(\hat{\mathcal{G}}) - \ln L(\mathcal{G}_i)}$ . We can further modify this formula to gain more control on performance of MCMC. Specifically, we introduce the *temperature*, or *tempering* parameter  $\tau$  as follows:

$$h(\hat{\mathcal{G}}, \mathcal{G}_i) = \exp\left(-\frac{\ln L(\mathcal{G}_i) - \ln L(\hat{\mathcal{G}})}{\tau}\right). \quad (15)$$

We note that this formula resembles the Boltzmann distribution, also known as Gibbs distribution, which is used in statistical and theoretical physics and describes probability to observe particles in the ground and excited energy states. Similarly, the parameter  $\tau$  controls how often MCMC accepts a network with worse likelihood than the current one, which in order affects the convergence of the algorithm and its ability to get out of local optima. If a new grid  $\hat{\mathcal{G}}$  has better likelihood than the current grid  $\mathcal{G}_i$ , it is always accepted. Otherwise, it is accepted only with probability  $p = L(\hat{\mathcal{G}})/L(\mathcal{G}_i) < 1$  for  $\tau = 1$ . For  $\tau > 1$ , this probability becomes  $p^{1/\tau} > p$ , which increases the probability that the new grid is accepted, encouraging MCMC to escape a local maximum.

### Choosing initial guess for optimisation

A good initial guess  $\mathcal{G}_0$  for the contact network can significantly improve the computational efficiency by reducing the number of steps required by the optimisation algorithm to converge towards the optimum  $\mathcal{G}_{\text{opt}}$ , but also by simplifying the forward problems and hence reducing the computational time required to evaluate each likelihood  $L(\mathcal{G})$  in (4). Here we present a simple algorithm to generate an initial guess using the given nodal

time series data  $\mathcal{X} = \{t_k, x_k\}_{k=0}^K$ . By comparing each next state  $x_k$  against a previous one  $x_{k-1}$  for  $k = 1, \dots, K$  node-by-node, we observe events of two possible types: infected nodes that become susceptible (recovery), and susceptible nodes becoming infected (infection). Recoveries are single-node events and provide no information on the network connectivity. In contrast, infections are two-node events that occur when a susceptible node is connected to an infected node. Therefore, any node  $m$  that was or became infected during  $t \in [t_{k-1}, t_k]$  could have infected any connected susceptible node  $n$  that became infected during the same time interval.

Thus, we compute the connectivity scores  $h_{m,n}$  for all  $m, n \in \mathcal{V}$  as follows

$$h_{m,n} = \sum_{k=1}^K h_{m,n}^{(k)}, \quad \text{with} \quad h_{m,n}^{(k)} = \begin{cases} \frac{1}{|\mathcal{I}_k|}, & \text{if } x_{n,k-1} = 0, x_{n,k} = 1, \text{ and } m \in \mathcal{I}_k; \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where  $\mathcal{I}_k = \{m \in \mathcal{V} : x_{m,k-1} = 1 \text{ or } x_{m,k} = 1\}$  is the set of all infected nodes at the beginning or by the end of the interval  $t \in [t_{k-1}, t_k]$ . The higher is the acquired score  $h_{m,n}$  the higher is the evidence that  $m \sim n$  in the contact network. Hence, when the scores are calculated, we can sample an initial guess network  $\mathcal{G}_0$  randomly with probabilities for each link proportional to the scores. Alternatively, for a more deterministic approach, we can discretise the distribution, and set  $m \sim n$  in  $\mathcal{G}_0$  for all links  $(m, n)$  for which the score exceeds the average,  $h_{m,n} \geq \frac{2}{N(N-1)} \sum_{i>j} h_{i,j}$ .

## Results

The numerical experiments were implemented in Matlab 2022b based on TT-Toolbox<sup>1</sup> and tAMEn<sup>2</sup> packages, and run on one node of the HC44 series of the University of Bath “Nimbus” Microsoft Azure cluster. Experiments with different datasets were run in parallel over 42 cores of the Intel Xeon Platinum 8168 CPUs. Each of these parallel processes ran in a single-threaded mode. The codes are made public and freely available from [github.com/savostyanov/ttsir](https://github.com/savostyanov/ttsir).

### Linear chain

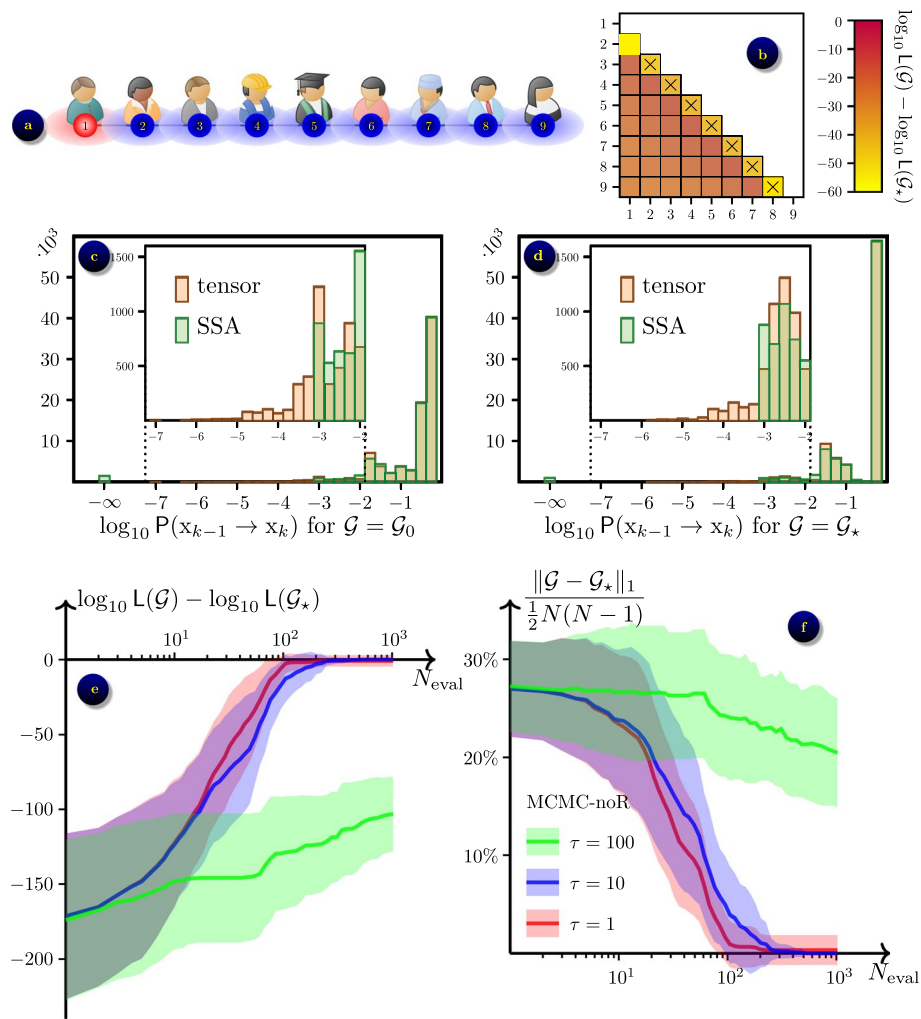
For this experiment we generated  $N_s = 42$  samples of synthetic data by computing random walks of  $\varepsilon$ -SIS process with parameters  $\beta = 1$ ,  $\gamma = 0.5$  and  $\varepsilon = 0.01$  for the duration of  $T = 200$  time units. The trajectories were then re-sampled to a uniform grid on  $[0, T]$  with the time step  $\Delta t = 0.1$  to imitate data collected at regular intervals. Therefore, each trajectory contained  $K = 2000$  data records representing the epidemic process. Data were created using the ‘ground truth’ network  $\mathcal{G}_\star$  which is a linear chain with  $N = 9$  nodes as shown in Fig. 5a, and assuming that the initial state  $x_0 = (1, 0, \dots, 0)$  is the same for all data samples.

First, we checked the contrast of the log-likelihood at the ground truth network  $\mathcal{G}_\star$ , by computing  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)]$  for all  $\mathcal{G}$  that are nearest neighbours of  $\mathcal{G}_\star$ , i.e. differ by only one link. The results are averaged over the  $N_s = 42$  sampled datasets and shown in Fig. 5b. We note that removal of an existing link from  $\mathcal{G}_\star$  results in contrast

<sup>1</sup> <https://github.com/oseledets/TT-Toolbox>.

<sup>2</sup> <https://github.com/dolgov/tamen>.





**Fig. 5** Inferring linear chain network with  $N = 9$  people from  $\varepsilon$ -SIS epidemic process with  $\beta = 1, \gamma = 0.5$  and  $\varepsilon = 0.01$ : **a** the ground truth network  $\mathcal{G}_*$  in its initial state; **b** the contrast  $\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)$  averaged over  $N_s = 42$  datasets, shown for grids  $\mathcal{G}$  that differ from  $\mathcal{G}_*$  by a single link  $(m, n)$ ; axes  $x$  show links in  $\mathcal{G}_*$ ; **c** the distribution of probabilities for the transitions observed in data for the initial guess network  $\mathcal{G}_0$ ; **d** the distribution of probabilities for the transitions observed in data for the ground truth network  $\mathcal{G}_*$ ; **e** convergence of likelihood  $L(\mathcal{G})$  towards  $L(\mathcal{G}_*)$  in the optimisation algorithm MCMC-noR; average (solid lines)  $\pm$  one standard deviation (shaded areas) over the  $N_s = 42$  datasets; shown for temperatures  $\tau = 1, 10, 100$ ; **f** convergence of network  $\mathcal{G}$  towards  $\mathcal{G}_*$ .

$E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)] \simeq -50$ , raising to  $\simeq -60$  for links attached to the sides of the chain. This is easy to understand, as removal of a link from  $\mathcal{G}_*$  creates a disconnected network  $\mathcal{G}$ , where two parts can not pass the infection on to each other, hence the epidemic dynamics on  $\mathcal{G}$  differs significantly from the one on  $\mathcal{G}_*$ . Adding a new link to  $\mathcal{G}_*$  results in a milder contrast  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)] \in [-30, -10]$ , because the grid remains connected and the dynamics of the epidemic is less affected. This confirms that  $\mathcal{G}_*$  is at least a local optimum for  $\log L(\mathcal{G})$ , and therefore can be inferred by Bayesian optimisation, assuming the optimisation algorithm manages to converge to it.

Secondly, we evaluated probabilities  $P(x_{k-1}^{(n_s)} \rightarrow x_k^{(n_s)} | \mathcal{G})$  in (4) for all data records  $k = 1, \dots, K$  for all generated datasets  $n_s = 1, \dots, N_s$ . The results are shown in Fig. 5c for

the initial guess network  $\mathcal{G} = \mathcal{G}_0$  computed as explained in Sect. “[Choosing initial guess for optimisation](#)”, and in Fig. 5d for the ground truth network  $\mathcal{G} = \mathcal{G}_*$ . We used the SSA algorithm [8] with  $N_{\text{SSA}} = 10^3$  samples as explained in Sect. “[Stochastic simulation algorithms for forward problem](#)”. A significant number of events are not resolved by SSA and the probabilities are estimated as zero, as shown by the  $\log_{10} p = -\infty$  column on the histograms. We then computed the same probabilities by solving the CME (11) subject to initial condition  $x_{k-1}^{(n_s)}$  on time interval  $t \in [t_{k-1}, t_k]$ , for which we apply the tAMEn algorithm [56] with Chebyshev polynomials of degree 12 in time and relative accuracy threshold  $\epsilon_{\text{tAMEn}} = 10^{-6}$ . From the tAMEn algorithm we obtain the whole p.d.f.  $\mathbf{p}(t) = [p(x, t)]_{x \in \mathbb{X}^N}$  for  $t \in [t_{k-1}, t_k]$  and for all states  $x \in \mathbb{X}^N$ , from which we extract the required probability by projecting to the deterministic final state  $x_k^{(n_s)}$ .

We observe that 1.7% of probabilities are unresolved by SSA for  $\mathcal{G} = \mathcal{G}_0$  and 1.1% of probabilities are unresolved for  $\mathcal{G} = \mathcal{G}_*$ , which is nevertheless sufficient for both likelihoods  $L(\mathcal{G}_0) = 0$  and  $L(\mathcal{G}_*) = 0$  to be unresolved for 100% of data samples  $n_s = 1, \dots, N_s$ .

The number of SSA trajectories is set to approximately match the computational time of SSA and tensor product algorithms for the forward problem. With tAMEn, the trajectories  $\mathbf{p}(t)$  are computed in the TT format (13) for which the TT ranks are determined adaptively. For this example we observe TT ranks  $r \simeq 14.8 \pm 2.3$  for  $\mathcal{G} = \mathcal{G}_0$  and  $r \simeq 11.1 \pm 0.9$  for  $\mathcal{G} = \mathcal{G}_*$  leading to computational time for the likelihood  $L(\mathcal{G})$  to be CPU time  $\simeq 98 \pm 6.9$  seconds for  $\mathcal{G} = \mathcal{G}_0$  and CPU time  $\simeq 80 \pm 3.2$  seconds for  $\mathcal{G} = \mathcal{G}_*$ . With the SSA algorithm, one likelihood computation took CPU time  $\simeq 199 \pm 23.5$  seconds for  $\mathcal{G} = \mathcal{G}_0$  and CPU time  $\simeq 107 \pm 6.3$  seconds for  $\mathcal{G} = \mathcal{G}_*$ . Note that the forward problems become easier to solve as the optimisation process approaches the ground truth network both because the linear geometry of the chain matches the structure of the TT format, and because the easier reaction network admits larger time steps in SSA. Due to the simplicity of the linear structure, a linear chain is an attractive model for study in quantum physics, see e.g a recent paper on the SIS model on a linear chain [35].

We performed the black-box Bayesian optimisation using  $N_{\text{eval}} = 10^3$  steps of the MCMC algorithms with and without replacement as explained in Sect. “[Algorithms for Bayesian inverse problem](#)”. We observed similar performance of both algorithms, hence only the results for MCMC-noR are shown in Fig. 5e for the convergence of the likelihood  $L(\mathcal{G})$  towards the one of the ground truth network,  $L(\mathcal{G}_*)$ , and in Fig. 5f for the corresponding convergence of the network  $\mathcal{G}$  towards the ground truth network  $\mathcal{G}_*$ . The latter is measured using the number of incorrectly inferred links,

$$\|\mathcal{G} - \mathcal{G}_*\|_1 = |\{m, n \in \mathcal{V}, m > n : g_{m,n} \neq g_{m,n}^*\}|, \quad (17)$$

related to the total number of possible links,  $\frac{1}{2}N(N-1)$ . For both MCMC-R and MCMC-noR without tempering (with  $\tau = 1$ ), we observe a steady convergence towards optimum with the ground truth grid correctly inferred in 40 out of 42 experiments and one link inferred incorrectly in 2 out of 42 experiments after  $N_{\text{eval}} = 10^3$  likelihood evaluations. To improve this result, we used tempering with temperature  $\tau = 10$ , and observed a slightly slower convergence of MCMC-noR, which then achieved the exact recovery for all  $N_s$  data samples. We also observed that increasing the temperature further to  $\tau = 100$  results in a much slower convergence and poor recovery, indicating that

this parameter needs to be carefully adjusted. In total for this experiment, the network inference from each dataset with  $K = 2000$  records took about  $7.3 \cdot 10^3$  seconds.

### Austria road network

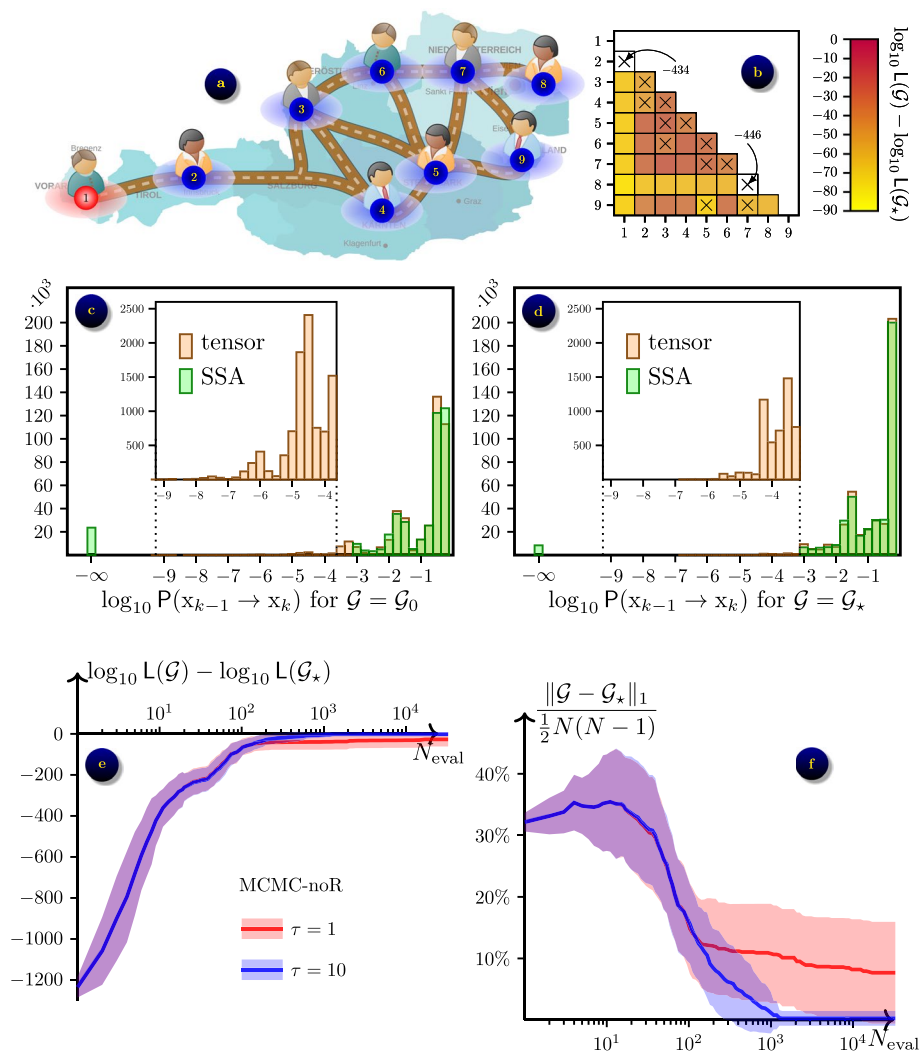
For this experiment we considered a more realistic example of a contact network, drawn from the road network in Austria, shown in Fig. 6a. As previously, we generated  $N_s = 42$  samples of synthetic data for a  $\varepsilon$ -SIS model with per contact transfer rate  $\beta = 1$ , individual recovery rate  $\gamma = 0.5$  and self-infection rate  $\varepsilon = 0.01$ . However, from preliminary experiments we noted that both MCMC algorithms for Bayesian optimisation struggle to converge to the optimum. To partly mitigate this, we increased the size of each dataset to  $K = 10^4$  data records, created by observing the state for the duration of  $T = 1000$  time units at uniform time grid with the step  $\Delta t = 0.1$ .

From the contrasts shown on Fig. 6b, we see that removal of any of two links that produces a disconnected graph  $\mathcal{G}$  results in a very high contrast,  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)] \lesssim -400$ . Removing or adding other links results in a connected  $\mathcal{G}$  and hence a moderate value of the contrast  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)] \in [-100, -15]$ .

Similarly to the previous example, we observe that SSA with  $n_{\text{SSA}} = 10^3$  samples does not resolve a significant number of events along the trajectory, and therefore returns  $\tilde{P}(x_{k-1} \rightarrow x_k | \mathcal{G}) = 0$  for 5.6% of data points for the initial guess network  $\mathcal{G} = \mathcal{G}_0$  and for 2.0% of data points for the ground truth network  $\mathcal{G} = \mathcal{G}_\star$ , leading to the likelihood  $L(\mathcal{G}) = 0$  being unresolved in all experiments for both grids. Note that the proportion of unresolved (rare) events is larger for this example due to a more complex network structure.

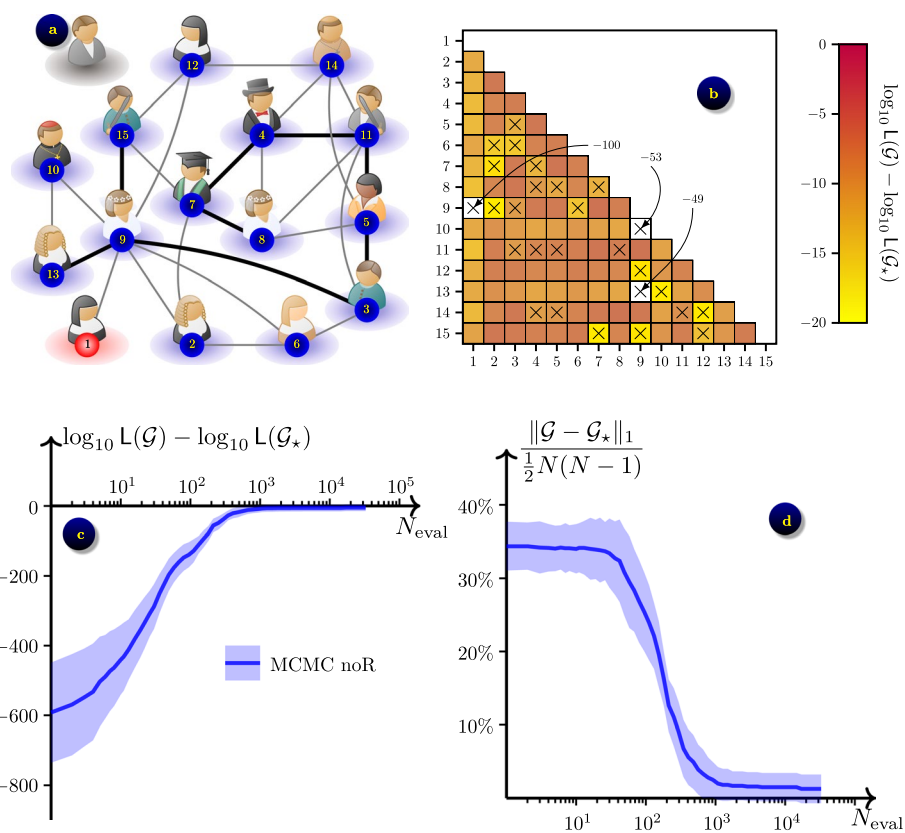
Using the tensor product approach with the same parameters as in Sect. “Linear chain” for the forward problem, we were able to resolve probabilities of up to  $p \sim 10^{-7}$ , which produced non-zero values for all likelihoods  $L(\mathcal{G})$ , enabling the optimisation for the inverse problem. For this example, one likelihood evaluation solving the forward problem with tAMEn took CPU time  $\simeq 431 \pm 8.2$  seconds for  $\mathcal{G} = \mathcal{G}_0$  and CPU time  $\simeq 439 \pm 4.8$  seconds for  $\mathcal{G} = \mathcal{G}_\star$ . The main reason for the larger times compared to the previous experiment is the larger data size  $K = 10^4$  compared to  $K = 2000$  for the linear chain. However, a more complex structure of the contact network also contributed via higher TT ranks  $r \simeq 12.0 \pm 1.8$  for  $\mathcal{G} = \mathcal{G}_0$  and  $r \simeq 13.4 \pm 1.2$  for  $\mathcal{G} = \mathcal{G}_\star$ . The total time required to perform the Bayesian optimisation with  $N_{\text{eval}} = 10^4$  steps of the MCMC algorithm took us about 45 hours. For comparison, when SSA is used as the forward solver, one likelihood computation took CPU time  $\simeq 1292 \pm 58.7$  seconds for  $\mathcal{G} = \mathcal{G}_0$  and CPU time  $\simeq 853 \pm 24.4$  seconds for  $\mathcal{G} = \mathcal{G}_\star$ .

From results shown in Fig. 6 we note that without tempering ( $\tau = 1$ ) the convergence of both MCMC algorithms is stuck in a local maximum where approximately 4 of 36 links are inferred incorrectly. Using tempering with  $\tau = 10$ , we observed almost the same convergence at initial stage of optimisation, which then resulted in a faster convergence towards a better inference, with 38 out of 42 data samples allowed for the exact recovery, and the remaining 4 out of 42 had only one incorrect link out of 36.



**Fig. 6** Inferring a road network in Austria ( $N = 9$  nodes) from  $\varepsilon$ -SIS epidemic process with  $\beta = 1, \gamma = 0.5$  and  $\varepsilon = 0.01$ : **a** the ground truth network  $\mathcal{G}_*$  in its initial state; **b** the contrast  $\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)$  averaged over  $N_s = 42$  datasets, shown for grids  $\mathcal{G}$  that differ from  $\mathcal{G}_*$  by a single link  $(m, n)$ ; axes  $x$  show links in  $\mathcal{G}_*$ ; **c** the distribution of probabilities for the transitions observed in data for the initial guess network  $\mathcal{G}_0$ ; **d** the distribution of probabilities for the transitions observed in data for the ground truth network  $\mathcal{G}_*$ ; **e** convergence of likelihood  $L(\mathcal{G})$  towards  $L(\mathcal{G}_*)$  in the optimisation algorithm MCMC-noR; average (solid lines)  $\pm$  one standard deviation (shaded areas) over the  $N_s = 42$  datasets; shown for temperatures  $\tau = 1, 10$ ; **f** convergence of network  $\mathcal{G}$  towards  $\mathcal{G}_*$ .

The difference in performance compared to the linear chain example can be considered as a consequence of a high contrast, that sharpens the high-dimensional landscape and makes both the global and local maxima steeper. If we use less data for Bayesian inference, the contrast reduces, making it easier for MCMC to escape from local optima by switching from current network  $\mathcal{G}_i$  to less attractive proposal  $\hat{\mathcal{G}}$  with probability  $L(\hat{\mathcal{G}})/L(\mathcal{G}_i) < 1$  as explained in Alg. 1. However, it also makes global maximum less emphasised and can lead to a situation where the optimal grid recovered by the Bayesian optimisation is not the same as the ground truth grid,  $\mathcal{G}_{opt} \neq \mathcal{G}_*$ .



**Fig. 7** Inferring a network of Florentine families ( $N = 15$  nodes) from  $\varepsilon$ -SIS epidemic process with  $\beta = 0.4, \gamma = 0.5$  and  $\varepsilon = 0.004$ : **a** the ground truth network  $\mathcal{G}_*$  in its initial state; **b** the contrast  $\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)$  averaged over  $N_s = 21$  datasets, shown for grids  $\mathcal{G}$  that differ from  $\mathcal{G}_*$  by a single link  $(m, n)$ ; axes  $x$  show links in  $\mathcal{G}_*$ ; **c** convergence of likelihood  $L(\mathcal{G})$  towards  $L(\mathcal{G}_*)$  in the optimisation process; average (solid lines)  $\pm$  one standard deviation (shaded areas) over the  $N_s = 21$  datasets; **d** convergence of network  $\mathcal{G}$  towards  $\mathcal{G}_*$ .

### Florentine families

We consider a slightly larger network representing marriage alliances and business relationships between Florentine families in XV century.<sup>3</sup> The network is given as an undirected weighted graph with 16 nodes, one of which is isolated from the rest, as shown in Fig. 7a. The weights of the links represent what kind of relationship the families have. For the purpose of this experiment we ignore the disconnected node and disregard the difference in connections. Hence we consider an undirected and unweighted network of  $N = 15$  nodes. Still, a more densely connected network leaves states more frequently in the infected state with  $\beta = 1$  used in the previous examples. To obtain a more meaningful data (and more accurate inference), we simulate the observation data using  $\beta = 0.4, \gamma = 0.5$ , and  $\varepsilon = 0.004$ . We observe states at uniformly distributed time points over the duration of  $T = 400$  time units, sampled with the time step  $\Delta t = 0.1$ , resulting in  $K = 4 \cdot 10^3$  data records.

From the contrasts shown in Fig. 7b we see that removal of the link between nodes 1 and 9 results in disconnected network, which results in a significant

<sup>3</sup> The network is taken from [networks.skewed.de/net/florentine\\_families](https://networks.skewed.de/net/florentine_families).

contrast,  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)] \lesssim -100$ . Notably, removal of the link (9, 10) does not fully disconnect the network, but considerably reduces the chance for the disease to reach from the node 9 to the node 13, hence the observed large contrast  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)] \approx -53$ . Similarly, removal of the link (9, 13) makes it harder for the disease to reach the node 10, which also results in a high contrast  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)] \approx -49$ . The remaining links seem to be considerably less important, and their removal results in a lower contrast  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_\star)] \gtrsim -20$ . On average, adding extra links also result in a lower contrast, with a notable exception of the first node, which is the origin of the epidemic. The lower contrast around the ground truth network may cause extra challenge for the exact recovery of this network, particularly if the likelihoods are computed inaccurately.

We run the MCMC-noR algorithm without tempering ( $\tau = 1$ ), and with tempering ( $\tau = 10$ ), but observe better results of the former. The convergence of the log-likelihood is shown in Fig. 7c, and the convergence of the inferred network is shown in Fig. 7d. We observe accurate recovery of the network, specifically, among the  $N_s = 21$  data samples that we tried, 11 resulted in exact recovery. In the remaining 10 data samples, we observed only 1 to 3 incorrectly recovered links. In our experiments, the MCMC-noR algorithm has reached the final network configuration after  $N_{\text{eval}} \approx 41 \cdot 10^3$  samples on average, which took 4.2 days of CPU time.

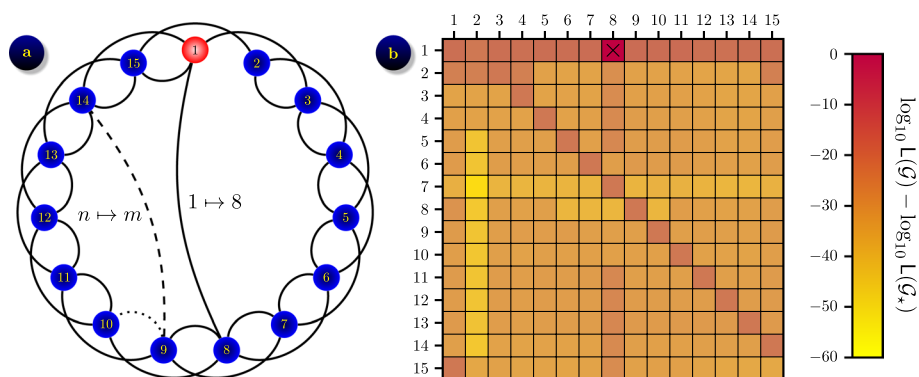
### Small world network

We note that even though the use of tensor product algorithms allows us to compute likelihoods (4) faster and more accurately, the exact Bayesian inference of a contact network in a fully black-box setting remains a challenging task, as we see from experiments in Sects. “Austria road network” and “Florentine families”.

In this section we present a preliminary experiment where we assume some prior knowledge of the contact network, which allows us to reduce the number of unknown parameters even for a larger number of network nodes. Specifically, we assume that the contact network is from a family of small-world networks [76], which is shown in Fig. 8a. It consists of  $N = 15$  nodes which are arranged as a loop and connected with a double bond, where each node  $n \in \mathcal{V}$  is connected to nodes  $n + 1$  and  $n + 2$ , where we assume that indices go around the circle, so  $N + 1 = 1$  and  $N + 2 = 2$  when necessary. The main loop is rewired, i.e a certain link  $(n, n + 1)$  is removed and replaced with a link  $(n, m)$  to a random node  $m \in \mathcal{V}$ , which provides additional connectivity. For this experiment we assumed that the ground truth network  $\mathcal{G}_\star$  contains a single rewired link  $1 \mapsto 8$ , i.e. the link  $(1, 2)$  is removed and replaced with  $(1, 8)$ . We then proceed to infer this network, assuming that we know it is from the set of small-world networks with a single rewired link  $n \mapsto m$ , which we denote  $\tilde{\mathcal{G}}$ . The problem therefore reduces to finding only two parameters,  $n$  and  $m$ , and the search space shrinks from  $|\mathcal{G}| = 2^{N(N-1)/2}$  to only  $|\tilde{\mathcal{G}}| = N^2$  possible grids.

Inferring a network from a known class can be formulated as Bayesian optimisation (4) on a class of networks  $\tilde{\mathcal{G}}$  parameterised by a small number of parameters. This removes our main computational challenge related to high dimensionality of the search space and allows us to solve this problem directly. We generated  $N_s = 42$  data samples by simulating the  $\varepsilon$ -SIS epidemic on a ground truth contact network  $\mathcal{G}_\star$  using parameters





**Fig. 8** Inferring a rewired link in small world graph ( $N = 15$  nodes) from  $\varepsilon$ -SIS epidemic process with  $\beta = 1, \gamma = 0.5$  and  $\varepsilon = 0.01$ : **a** the ground truth grid  $\mathcal{G}_*$  shown in its initial state; **b** the contrast  $\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)$  averaged over  $N_s = 42$  datasets, shown for grids  $\mathcal{G} \in \tilde{\mathcal{G}}$  from a class of small-world networks with a single rewired link

$\beta = 1, \gamma = 0.5$  and  $\varepsilon = 0.01$ , for the duration of  $T = 1000$  time units, and re-sampled the data to a uniform grid with the time step  $\Delta t = 0.1$ , hence creating  $K = 10^4$  records for each data sample. Using tAMEn algorithm to model the evolution of epidemic on 15-node networks, we were able to compute the likelihoods for all grids  $\mathcal{G} \in \tilde{\mathcal{G}}$ . We then computed the average contrast for all  $\mathcal{G} \in \tilde{\mathcal{G}}$  as shown in Fig. 8b. The results show that  $E[\log_{10} L(\mathcal{G}) - \log_{10} L(\mathcal{G}_*)] \leq -10$  for all  $\mathcal{G} \neq \mathcal{G}_*$ , which ensures that the ground truth network is a unique global maximum of the Bayesian optimisation problem (4).

### Discussion

Inferring the contact network in a Bayesian optimisation framework requires us to estimate the likelihood of observed data  $\mathcal{X}$ , which are a realisation of epidemic dynamics on the ground truth network  $\mathcal{G}_*$ , to appear for the epidemic on another network  $\mathcal{G}$ . In a black-box setting, we have no a priori information on the network, and start the optimisation from an initial guess  $\mathcal{G}_0$  that may be (very) different from  $\mathcal{G}_*$ . For the grids  $\mathcal{G}$  in the vicinity of  $\mathcal{G}_0$ , observing the same dynamics as on  $\mathcal{G}_*$  is a (very) rare event, which we need to estimate with sufficient precision in order to evaluate the likelihoods  $L(\mathcal{G})$ . The slow convergence of the SSA algorithm limits its capability to recover rare events. By replacing it with the tensor product algorithms, we are able to recover rare events much more accurately by solving the forward problem in the CME form (11) and overcoming the curse of dimensionality. This allows the MCMC method to find its way from the initial network  $\mathcal{G}_0$  towards the optimum.

As the optimisation gets closer to  $\mathcal{G}_*$ , the likelihoods increase and the presence of steep local maxima slows down the convergence towards the global one. In this area high contrast ratios  $L(\mathcal{G}_*)/L(\mathcal{G})$  are undesirable as they make it harder for the MCMC algorithm to escape local maxima. By preliminary experiments demonstrated in this paper we show that this can be addressed tempering of  $L(\mathcal{G})$  to simplify the high-dimensional landscape for the optimisation. Another idea is to use only a part of the available data to compute likelihoods (4), which has been used successfully for sampling from concentrated distributions of continuous random variables [77].

We also explored the potential of tensor product algorithms for tackling the network likelihood optimisation. However, these attempts so far were less efficient than the MCMC algorithm (in particular the MCMC without replacement). The TT-Cross algorithm [78] and its extensions [79] are used to compute a TT approximation to a black-box tensor by drawing a few adaptive samples from it using the maximum volume algorithm [80] or a greedy version thereof [79]. These maximum volume samples are expected to be good candidates for the maximum absolute value of the tensor [80, 81]. However, the maximum volume algorithm requires all elements of a TT core, which must be drawn as full columns from the tensor, including elements which are known to be far from the maximum. MCMC probes only one element at a time, and can skip such unnecessary calculations. In numerical experiments with the linear chain, MCMC was systematically faster and more accurate compared to the TT-Cross maximiser, albeit by a modest margin (1–2 contacts). Tempering the likelihood to reduce its TT ranks and caching its values (which are often repeated in the TT-Cross) may make this approach faster in terms of the actual CPU time.

Another tensor optimiser proposed recently is PROTES [82], a probabilistic method similar to genetic algorithms. In each iteration, this algorithm draws  $N_c$  candidate optima as random samples from a probability distribution function in the TT format, which is in turn updated by a stochastic gradient ascent maximising the probability of drawing  $n_s$  samples with the largest values of the sought function out of the  $N_c$  candidates. The default parameters proposed in [82] are  $n_s = 10$  and  $N_c = 100$ . Compared to our budget of  $N_{\text{eval}} = 400$  function evaluations, this corresponds to only 4 stochastic gradient ascent iterations, which are clearly insufficient and produce an almost random network. Taking  $N_c$  in the order of 10 (and hence  $n_s < 10$ ) is uncompetitive too, since a few tens of iterations cannot compensate for a more random stochastic gradient due to a smaller  $n_s$ . However, it may be reasonable to use such an algorithm to fine-tune a previous TT approximation of the likelihood to new data.

Choosing a more informative prior on the network may aid the inference. We have already stepped away from a fully uniform prior in the small world example, where we sought only one rewiring instead of the state of all links. Penalising improbable or redundant links with a low prior probability may be beneficial for more general networks as well.

Potentially, it may be possible to use all MCMC points to compute posterior expectations rather than the MLE/MAP. However, this would be difficult for network identification for two reasons. First, accurate sampling would need much more likelihood evaluations (and hence CPU time) to decorrelate the Markov chain, whereas the MLE/MAP can be found in a few thousand samples. Secondly, the expected state would be a real-valued instead of binary vector, and require ambiguous post-processing to convert it into a network. Mitigation of these obstacles can be a matter of a future research.

#### Abbreviations

SIR	Susceptible-infected-recovered epidemic model, see [1]
SIS	Susceptible-infected-susceptible epidemic model, see e.g. [83]
ODE	Ordinary differential equation
CME	Chemical master equation, see [7]
MLE	Maximum likelihood estimate, see e.g. [84]
MAP	Maximum a posteriori estimate, see e.g. [84]

SSA	Stochastic simulation algorithm, see [8]
MCMC	Markov chain Monte Carlo [74]
MCMC-noR	A version of MCMC without replacement, see Sect. "Algorithms for Bayesian inverse problem"
CP	Canonical polyadic tensor product format, see [50–52]
TT	Tensor train format, see [64]
MPS	Matrix product states, see e.g. [73]
DMRG	Density matrix renormalisation group algorithm, see [85]
AMEn	Alternating minimal energy method, see [55]
tAMEn	Time-dependent AMEn, see [56]
CPU time	Central processing unit time, also known as the wallclock time

#### Author contributions

Sergey Dolgov developed software, performed numerical experiments, analysed the results, and contributed to writing the manuscript. Dmitry Savostyanov designed the work, analysed the results, designed visualisations, and was a major contributor to writing the manuscript. Both authors read and approved the final manuscript.

#### Funding

Sergey Dolgov was supported by the Engineering and Physical Sciences Research Council (EPSRC) New Investigator Award EP/T031255/1. Dmitry Savostyanov was supported by the Leverhulme Trust Research Fellowship RF-2021-258 at the initial stage of this work. Dmitry Savostyanov would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme Discretization and recovery in high-dimensional spaces, where work on the revised version of this paper was undertaken. This work was partly supported by EPSRC grant no EP/R014604/1 and by a grant from the Simons Foundation. Funders took no part in study design; in the collection, analysis and interpretation of data; in the writing of the report; and in the decision to submit the article for publication.

#### Data and code availability

Numerical experiments in this work are based on synthetic randomly generated datasets. All data and code required to reproduce experiments are available on [github.com/savostyanov/ttsir](https://github.com/savostyanov/ttsir).

#### Materials availability

Not applicable.

#### Declarations

##### Competing interests

Authors declare no conflict of interest exist.

##### Ethics approval and consent to participate

Not applicable.

##### Consent for publication

Not applicable.

Received: 29 January 2024 Accepted: 21 August 2024

Published online: 02 September 2024

#### References

- Kermack WO, McKendrick AG. A contribution to the mathematical theory of epidemics. *Proc R Soc Lond A*. 1927;115(772):700–21. <https://doi.org/10.1098/rspa.1927.0118>.
- Keeling MJ, Eames KTD. Networks and epidemic models. *Interface*. 2005;2(4):295–307. <https://doi.org/10.1098/rsif.2005.0051>.
- Chen W-Y, Bokka S. Stochastic modeling of nonlinear epidemiology. *J Theor Biol*. 2005;234(4):455–70. <https://doi.org/10.1016/j.jtbi.2004.11.033>.
- Youssef M, Scoglio C. An individual-based approach to SIR epidemics in contact networks. *J Theor Biol*. 2011;283(1):136–44. <https://doi.org/10.1016/j.jtbi.2011.05.029>.
- Pastor-Satorras R, Castellano C, Van Mieghem P, Vespignani A. Epidemic processes in complex networks. *Rev Mod Phys*. 2015;87:925. <https://doi.org/10.1103/RevModPhys.87.925>.
- Kiss IZ, Miller JC, Simon PL. *Mathematics of epidemics on networks: from exact to approximate models*. Berlin: Springer; 2017.
- Kampen NG. *Stochastic processes in physics and chemistry*. Amsterdam: North Holland; 1981.
- Gillespie DT. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J Comput Phys*. 1976;22(4):403–34. [https://doi.org/10.1016/0021-9991\(76\)90041-3](https://doi.org/10.1016/0021-9991(76)90041-3).
- Keeling MJ. The effects of local spatial structure on epidemiological invasions. *Proc Biol Sci*. 1999;266(1421):859–67. <https://doi.org/10.1098/rspb.1999.0716>.
- Rand DA. Correlation equations and pair approximations for spatial ecologies. In: *Advanced ecological theory: principles and applications*, pp. 100–142. Blackwell Science, Oxford; 1999.
- Gleeson JP. High-accuracy approximation of binary-state dynamics on networks. *Phys Rev Lett*. 2011;107: 068701. <https://doi.org/10.1103/PhysRevLett.107.068701>.

12. Lindquist J, Ma J, Driessche P, Willeboordse FH. Effective degree network disease models. *J Math Biol.* 2011;62:143–64. <https://doi.org/10.1007/s00285-010-0331-2>.
13. Taylor M, Taylor TJ, Kiss IZ. Epidemic threshold and control in a dynamic network. *Phys Rev E.* 2012;85: 016103. <https://doi.org/10.1103/PhysRevE.85.016103>.
14. Miller JC, Slim AC, Volz EM. Edge-based compartmental modelling for infectious disease spread. *J R Soc Interface.* 2012;9:890–906. <https://doi.org/10.1098/rsif.2011.0403>.
15. Di Lauro F, Croix J-C, Dashti M, Berthouze L, Kiss IZ. Network inference from population-level observation of epidemics. *Sci Rep.* 2020;10:18779. <https://doi.org/10.1038/s41598-020-75558-9>.
16. Gupta A, Schwab C, Khammash M. DeepCME: a deep learning framework for computing solution statistics of the chemical master equation. *PLoS Comput Biol.* 2021;17(12):1009623. <https://doi.org/10.1371/journal.pcbi.1009623>.
17. Sukys A, Öcal K, Grima R. Approximating solutions of the chemical master equation using neural networks. *iScience.* 2022;25: 105010. <https://doi.org/10.1016/j.isci.2022.105010>.
18. Dolgov S, Savostyanov D. Tensor product approach to modelling epidemics on networks. *Appl Math Comput.* 2024;460: 128290. <https://doi.org/10.1016/j.amc.2023.128290>.
19. De Smet R, Marchal K. Advantages and limitations of current network inference methods. *Nature Rev Microbiol.* 2010;8:717–29. <https://doi.org/10.1038/nrmicro2419>.
20. O'Neill PD, Roberts GO. Bayesian inference for partially observed stochastic epidemics. *J R Stat Soc A.* 1999;162(1):121–9. <https://doi.org/10.1111/1467-985X.00125>.
21. Economou A, Gómez-Corral A, López-García M. A stochastic SIS epidemic model with heterogeneous contacts. *Physica A.* 2015;421:78–97. <https://doi.org/10.1016/j.physa.2014.10.054>.
22. Keeling MJ, Rohani P. Estimating spatial coupling in epidemiological systems: a mechanistic approach. *Ecol Lett.* 2002;5(1):20–9. <https://doi.org/10.1046/j.1461-0248.2002.00268.x>.
23. Enciso G, Erban R, Kim J. Identifiability of stochastically modelled reaction networks. *Eur J Appl Math.* 2021;32(5):865–87. <https://doi.org/10.1017/S0956792520000492>.
24. Mukherjee S, Speed TP. Network inference using informative priors. *Proc Natl Acad Sci.* 2008;105(38):14313–8. <https://doi.org/10.1073/pnas.0802272105>.
25. Han X, Shen Z, Wang W-X, Di Z. Robust reconstruction of complex networks from sparse data. *Phys Rev Lett.* 2015;114: 028701. <https://doi.org/10.1103/PhysRevLett.114.028701>.
26. Peixoto TP. Network reconstruction and community detection from dynamics. *Phys Rev Lett.* 2019;123: 128301. <https://doi.org/10.1103/PhysRevLett.123.128301>.
27. Groendyke C, Welch D, Hunter DR. Bayesian inference for contact networks given epidemic data. *Scand J Stat.* 2011;38(3):600–16. <https://doi.org/10.1111/j.1467-9469.2010.00721.x>.
28. Landry NW, Thompson W, Hébert-Dufresne L, Young J-G. Complex contagions can outperform simple contagions for network reconstruction with dense networks or saturated dynamics. 2024. <https://arxiv.org/pdf/2405.00129>
29. Murphy C, Thibeault V, Allard A, Desrosiers P. Duality between predictability and reconstructability in complex systems. *Nature Commun.* 2024;15:4478. <https://doi.org/10.1038/s41467-024-48020-x>.
30. Shandilya SG, Timme M. Inferring network topology from complex dynamics. *New J Phys.* 2011;13(1): 013004. <https://doi.org/10.1088/1367-2630/13/1/013004>.
31. Britton T, Trapman P. Inferring global network properties from egocentric data with applications to epidemics. *Math Med Biol: J IMA.* 2015;32(1):101–14. <https://doi.org/10.1093/imammb/dqt022>.
32. Lokhov AY, Mézard M, Ohta H, Zdeborová L. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Phys Rev E.* 2014;90: 012801. <https://doi.org/10.1103/PhysRevE.90.012801>.
33. Brugere I, Gallagher B, Berger-Wolf TY. Network structure inference, a survey: motivations, methods, and applications. *ACM Comput Surv.* 2018;51(2):24–12439.
34. Ma C, Chen H-S, Lai Y-C, Zhang H-F. Statistical inference approach to structural reconstruction of complex networks from binary time series. *Phys Rev E.* 2018;97: 022301. <https://doi.org/10.1103/PhysRevE.97.022301>.
35. Merbis W, Mulatier C, Corboz P. Efficient simulations of epidemic models with tensor networks: application to the one-dimensional susceptible-infected-susceptible model. *Phys Rev E.* 2023;108: 024303. <https://doi.org/10.1103/PhysRevE.108.024303>.
36. Hill AL, Rand DG, Nowak MA, Christakis NA. Emotions as infectious diseases in a large social network: the SISa model. *Proc R. Soc B.* 277(1701) (2010). <https://doi.org/10.1098/rspb.2010.1217>
37. Van Mieghem P, Cator E. Epidemics in networks with nodal self-infection and the epidemic threshold. *Phys Rev E.* 2012;86: 016116. <https://doi.org/10.1103/PhysRevE.86.016116>.
38. Achterberg MA, Prasse B, Van Mieghem P. Analysis of continuous-time markovian  $\epsilon$ -SIS epidemics on networks. *Phys Rev E.* 2022;105: 054305. <https://doi.org/10.1103/PhysRevE.105.054305>.
39. Zhang J, Moura JMF, Zhang J. Contact process with exogenous infection and the scaled SIS process. *J Complex Netw.* 2017;5(5):712–33. <https://doi.org/10.1093/comnet/cnx003>.
40. Box GEP, Tiao GC. Bayesian inference in statistical analysis. NY: Wiley; 1973.
41. Gillespie DT. Approximate accelerated stochastic simulation of chemically reacting systems. *J Chem Phys.* 2001;115(4):1716–33. <https://doi.org/10.1063/1.1378322>.
42. Hemberg M, Barahona M. Perfect sampling of the master equation for gene regulatory networks. *Biophys J.* 2007;93(2):401–10. <https://doi.org/10.1529/biophysj.106.099390>.
43. Anderson DF, Higham DJ. Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics. *Multiscale Model Simul.* 2012;10(1):146–79. <https://doi.org/10.1137/110840546>.
44. Lester C, Baker RE, Giles MB, Yates CA. Extending the multi-level method for the simulation of stochastic biological systems. *Bull Math Biol.* 2016;78(8):1640–77. <https://doi.org/10.1007/s11538-016-0178-9>.
45. Hegland M, Burden C, Santoso L, MacNamara S, Booth H. A solver for the stochastic master equation applied to gene regulatory networks. *J Comput Appl Math.* 2007;205(2):708–24. <https://doi.org/10.1016/j.cam.2006.02.053>.
46. Minsky B, Khammash M. The finite state projection algorithm for the solution of the chemical master equation. *J Chem Phys.* 2006;124: 044104. <https://doi.org/10.1063/1.2145882>.

47. Jahnke T. An adaptive wavelet method for the chemical master equation. *SIAM J Sci Comput.* 2010;31(6):4373. <https://doi.org/10.1137/080742324>.
48. Cao Y, Terebus A, Liang J. State space truncation with quantified errors for accurate solutions to discrete chemical master equation. *Bull Math Biol.* 2016;78(4):617–61. <https://doi.org/10.1007/s11538-016-0149-1>.
49. Kryven I, Röblitz S, Schütte C. Solution of the chemical master equation by radial basis functions approximation with interface tracking. *BMC Syst Biol.* 2015;9(1):67. <https://doi.org/10.1186/s12918-015-0210-y>.
50. Jahnke T, Huisinga W. A dynamical low-rank approach to the chemical master equation. *Bull Math Biol.* 2008;70:2283–302. <https://doi.org/10.1007/s11538-008-9346-x>.
51. Ammar A, Cueto E, Chinesta F. Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions. *Int J Numer Methods Biomed Eng.* 2012;28(9):960–73. <https://doi.org/10.1002/cnm.2476>.
52. Hegland M, Garcke J. On the numerical solution of the chemical master equation with sums of rank one tensors. *ANZIAM* 2011;52:628–643. <https://doi.org/10.21914/anziamj.v52i0.3895>
53. Kazeev V, Khammash M, Nip M, Schwab C. Direct solution of the chemical master equation using quantized tensor trains. *PLoS Comput Biol.* 2014;10(3): 100359. <https://doi.org/10.1371/journal.pcbi.1003359>.
54. Dolgov S, Khromskij B. Simultaneous state-time approximation of the chemical master equation using tensor product formats. *Numer Linear Algebra Appl.* 2015;22(2):197–219. <https://doi.org/10.1002/nla.1942>.
55. Dolgov SV, Savostyanov DV. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM J Sci Comput.* 2014;36(5):2248–71. <https://doi.org/10.1137/140953289>.
56. Dolgov SV. A tensor decomposition algorithm for large ODEs with conservation laws. *Comput Methods Appl Math.* 2019;19:23–38. <https://doi.org/10.1515/cmam-2018-0023>
57. Vo HD, Sidje RB. An adaptive solution to the chemical master equation using tensors. *J Chem Phys.* 2017;147(4): 044102. <https://doi.org/10.1063/1.4994917>.
58. Dinh T, Sidje RB. An adaptive solution to the chemical master equation using quantized tensor trains with sliding windows. *Phys Biol.* 2020;17(6): 065014. <https://doi.org/10.1088/1478-3975/aba1d2>.
59. Ion IG, Wildner C, Loukrezis D, Koepl H, De Gersem H. Tensor-train approximation of the chemical master equation and its application for parameter inference. *J Chem Phys.* 2021;155(3): 034102. <https://doi.org/10.1063/5.0045521>.
60. Gellß P, Matera S, Schütte C. Solving the master equation without kinetic Monte Carlo: Tensor train approximations for a CO oxidation model. *J Comput Phys.* 2016;314:489–502. <https://doi.org/10.1016/j.jcp.2016.03.025>.
61. Hitchcock FL. The expression of a tensor or a polyadic as a sum of products. *J Math Phys.* 1927;6(1):164–89.
62. Harshman RA. Foundations of the PARAFAC procedure: models and conditions for an explanatory multimodal factor analysis. In: *UCLA Working Papers in Phonetics* 1970;16:1–84
63. Carroll JD, Chang JJ. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition. *Psychometrika.* 1970;35:283–319. <https://doi.org/10.1007/BF02310791>.
64. Oseledets IV. Tensor-train decomposition. *SIAM J Sci Comput.* 2011;33(5):2295–317. <https://doi.org/10.1137/090752286>.
65. Rohrbach PB, Dolgov S, Grasedyck L, Scheichl R. Rank bounds for approximating Gaussian densities in the Tensor-Train format. *SIAM/ASA J Uncertain Quant.* 2022;10(3):1191–224. <https://doi.org/10.1137/20M1314653>.
66. Ballani J, Grasedyck L. Tree adaptive approximation in the hierarchical tensor format. *SIAM J Sci Comput.* 2014;36(4):1415–31. <https://doi.org/10.1137/130926328>.
67. Michel B, Nouy A. Learning with tree tensor networks: complexity estimates and model selection. *Bernoulli.* 2022;28(2):910–36. <https://doi.org/10.3150/21-BEJ1371>.
68. Cui T, Dolgov S, Zahm O. Scalable conditional deep inverse rosenblatt transports using tensor trains and gradient-based dimension reduction. *J Comput Phys.* 2023;485: 112103. <https://doi.org/10.1016/j.jcp.2023.112103>.
69. Barcza G, Legeza O, Marti KH, Reiher M. Quantum-information analysis of electronic states of different molecular structures. *Phys Rev A.* 2011;83: 012508. <https://doi.org/10.1103/PhysRevA.83.012508>.
70. Fiedler M. Algebraic connectivity of graphs. *Czechoslovak Math J* 1973;23(2):298–305. <https://doi.org/10.21136/CMJ.1973.101168>
71. Fiedler M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Math J.* 1975, 25(4), pp. 619–633. <https://doi.org/10.21136/CMJ.1975.101357>
72. Barnard ST, Pothan A, Simon HD. A spectral algorithm for envelope reduction of sparse matrices. In: *Proceedings of ACM/IEEE conference on supercomputing*, 1993;493–502. <https://doi.org/10.1109/SUPERC.1993.1263497>
73. Schollwöck U. The density-matrix renormalization group in the age of matrix product states. *Ann Phys.* 2011;326(1):96–192. <https://doi.org/10.1016/j.aop.2010.09.012>.
74. Roberts GO, Rosenthal JS. Quantitative non-geometric convergence bounds for independence samplers. *Methodol Comput Appl Probab.* 2011;13(2):391–403.
75. Shah R, Kroese DP. Without-replacement sampling for particle methods on finite state spaces. *Stat Comput.* 2018;28(3):633–52. <https://doi.org/10.1007/s11222-017-9752-8>.
76. Watts DJ, Strogatz SH. Collective dynamics of ‘small-world’ networks. *Nature.* 1998;393:440–2. <https://doi.org/10.1038/30918>.
77. Cui T, Dolgov S. Deep composition of Tensor-Trains using squared inverse Rosenblatt transports. *Found Comput Math.* 2022;22(6):1863–922. <https://doi.org/10.1007/s10208-021-09537-5>.
78. Oseledets IV, Tyrtyshnikov EE. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.* 2010;432(1):70–88. <https://doi.org/10.1016/j.laa.2009.07.024>.
79. Dolgov S, Savostyanov D. Parallel cross interpolation for high-precision calculation of high-dimensional integrals. *Comput Phys Commun.* 2020;246:106869. <https://doi.org/10.1016/j.cpc.2019.106869>
80. Goreinov SA, Oseledets IV, Savostyanov DV, Tyrtyshnikov EE, Zamarashkin NL. How to find a good submatrix. In: Olshevsky V, Tyrtyshnikov E, editors. *Matrix methods: theory, algorithms, applications*. Hackensack, NY: World Scientific; 2010. p. 247–56.
81. Sozykin K, Chertkov A, Schutski R, Phan A-H, Cichocki AS, Oseledets I. TTOPt: a maximum volume quantized tensor train-based optimization and its application to reinforcement learning. In: *Advances in neural information processing systems*, 2022;35:26052–26065. [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/a730abbc6cf4a371ca9545db5922442-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/a730abbc6cf4a371ca9545db5922442-Paper-Conference.pdf)

82. Batsheva A, Chertkov A, Ryzhakov G, Oseledets I. PROTES: probabilistic optimization with tensor sampling. 2023. <http://arxiv.org/abs/2301.12162>
83. Van Mieghem P, Omic J, Kooij R. Virus spread in networks. *IEEE/ACM Trans Netw.* 2009;17(1):1–14. <https://doi.org/10.1109/TNET.2008.925623>.
84. Bain LJ, Engelhardt M. *Introduction to probability and mathematical statistics*. Belmont: Duxbury Press; 1992.
85. White SR. Density-matrix algorithms for quantum renormalization groups. *Phys Rev B.* 1993;48(14):10345–56. <https://doi.org/10.1103/PhysRevB.48.10345>.

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.