



CHORUS

This is the accepted manuscript made available via CHORUS. The article has been published as:

Detecting hidden layers from spreading dynamics on complex networks

Łukasz G. Gajewski, Jan Chołoniowski, and Mateusz Wilinski

Phys. Rev. E **104**, 024309 — Published 11 August 2021

DOI: [10.1103/PhysRevE.104.024309](https://doi.org/10.1103/PhysRevE.104.024309)

Detecting Hidden Layers from Spreading Dynamics on Complex Networks

Łukasz G. Gajewski* and Jan Chołoniowski

*Center of Excellence for Complex Systems Research, Faculty of Physics,
Warsaw University of Technology, Koszykowa 75, 00-662, Warsaw, Poland*

Mateusz Wilinski

Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

When dealing with spreading processes on networks it can be of the utmost importance to test the reliability of data and identify potential unobserved spreading paths. In this paper we address these problems and propose methods for hidden layer identification and reconstruction. We also explore the interplay between difficulty of the task and the structure of the multilayer network describing the whole system where the spreading process occurs. Our methods stem from an exact expression for the likelihood of a cascade in the Susceptible-Infected model on an arbitrary graph. We then show that by imploring statistical properties of unimodal distributions and simple heuristics describing joint likelihood of a series of cascades one can obtain an estimate of both existence of a hidden layer and its content with success rates far exceeding those of a null model. We conduct our analyses on both synthetic and real-world networks providing evidence for the viability of the approach presented.

I. INTRODUCTION

Real-world complex systems can often be described by interconnected structures known as multilayer networks [1–4]. Transportation, social or economic networks, to name just a few general examples, can have various types of connections, see Fig. 1 for an example depiction of such a system. Each such type of a connection in a network can be represented as a specific sub-system or sub-network. Railway, flights and bus connections can all be described with a network but to have a full description of the transportation system, they need to be joined and described with a multilayer network. In reality, obtaining full information which would allow to create a complete multilayer network is rarely possible. Moreover, in some cases, even the knowledge about all existing layers is limited. As a result, researchers often have to deal with uncertainty which arise from dealing with partial information about connectivity in analysed system. This specifically concerns one of the fundamental problems in network science, the spreading processes on networks [5–11] but is also of significance for opinion dynamics [12–15].

In the following article we focus on the problem of detecting hidden layers based on observations of a dynamical processes on graphs. By dynamical process we mean a realisation of a spreading process described by a model of the susceptible-infected-recovered (SIR) type. Note that such models can describe not only infections, but also spreading of information, opinions or failures. Furthermore, we assume that the observation of such a process is limited to the states of the nodes, without the knowledge of the actual spreading path. In the rest of the text we will refer to a single spreading realisation as a cascade. We also propose and explore methods for finding missing connections of different types. Finally, we analyse

potential limitations and difficulties as well as beneficial settings, i.e. when solving the problem is easier, for these methods.

The problem of detecting hidden layers has appeared recently in the literature in a non-markovian setting [16] and on quantum graphs [17]. It is also closely related to the problem of network reconstruction which was extensively analysed in the past [18–22] and also in a partial observation setting [23–25]. Our setting is a bit simpler in some regards but at the same time still fairly realistic and thus should still be viable for real-world problems. We feel that our simplifications are justified since solving the general problem was proved to be limited [19] and previous papers often approached only limited cases anyway, such as very short cascades [26]. This is not to say that successful approximations are not possible [18], however, the goal of this paper is to investigate the challenges associated with detecting hidden layers in interconnected networks in the context of spreading processes.

Reader should not confuse the problem of finding hidden layers based on observed spreading with extensively analysed branch of network science called *link prediction*, where the hidden connections are estimated using only the network structure. A seminal paper in this direction is [27]. An extension, including multilayer networks, can be found in [28].

The paper is structured as follows: in the next section we describe all the methods used in the analysis, starting with tools which allow for detecting hidden layers and then proposing methods for identifying unobserved connections. In the third section we analyse both synthetic and real world networks and show how our methods work under different structural circumstances. Finally, we discuss all the results and present our conclusions in the last section.

* lukaszgajewski@tuta.io

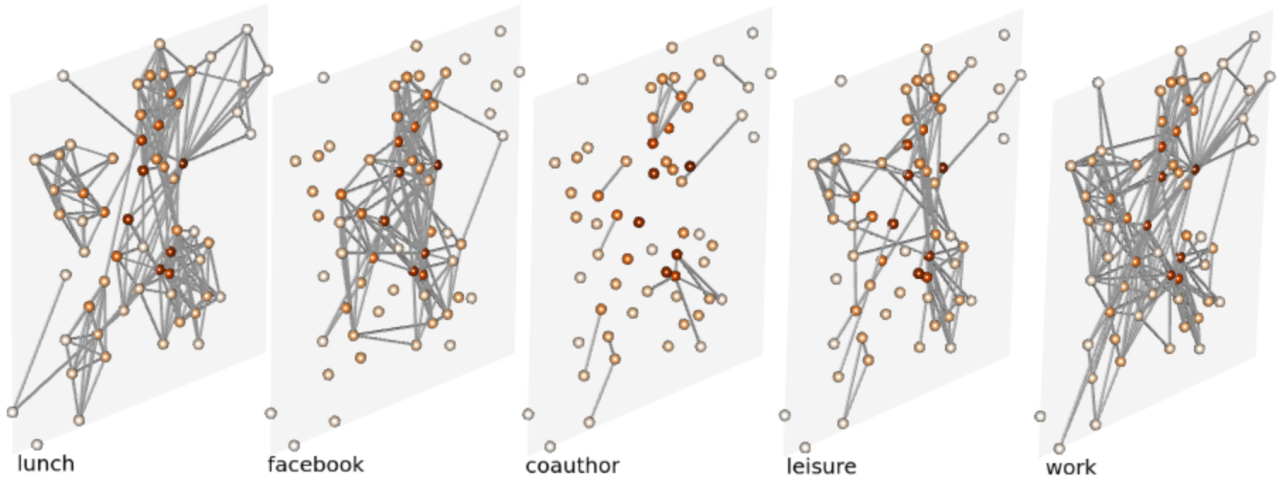


FIG. 1. Visualisation of the multilayer network representing the Aarhus data [29]. We utilise this network as a real world example of possible application of our methods. It is quite natural to imagine we know only one of the layers presented here and would like to infer the existence and possibly structure of others. Note that co-author and leisure layers have lower connections' density in comparison with others. Thus we do not use them as visible layers in our analyses as that makes the task of detecting hidden connections potentially much easier.

II. METHODS

In our analysis we focus on one of the simplest spreading model – the Susceptible Infected (SI) model [30]. We use it in a network version where the dynamics can be described as follows: for each node i , which is in the infected state I at time t , each of its neighbors j (in susceptible state S) will become infected at time $t + 1$ with probability β . This model was used because of its simplicity on one hand and the mechanism of multiple infection opportunities (in comparison with, e.g. an Independent Cascade model) on the other. The latter makes the combinatorial analysis much more difficult, as we will see in further sections. The former is reflected by an easy to derive likelihood of any observed spreading, including a multilayer scenario, assuming that the full knowledge about connectivity is available. We are well aware that for specific applications other, often more complex, models may be better suited for the problem. Our approach, however, can effectively be used for other spreading description. As an example, we provide the full derivation and results also for the Independent Cascade model (see Appendix A). Note that SI and IC are limiting versions of the SIR model, which means that the presented results give limits for the whole family of SIR models.

Let us denote a multilayer graph with G and let the probability of infection (spreading) on each layer j be equal to β_j . We will refer to a single spreading dynamics as a cascade and denote it with Σ^c . A single cascade can be described by a set of infection times τ_i^c for each node i . We will also assume that a cascade ends at time t_{max} and if a given node was not infected at all, its infection time will be equal to t_{max} . In other words, if node's i activation times is equal to t_{max} , it was either activated at t_{max}

or later – this will be more clear once the likelihood is derived. The set of all available cascades will be denoted with Σ .

A. Cascade likelihood for the Susceptible-Infected model

As mentioned before, the likelihood of a given set of cascades, for a specific and fully known multilayer network can be derived, similarly as it was done in [23]. In short, the probability of a given data-set can be written as a product over cascades, which are independent, and nodes, because the problem can be considered locally:

$$P(\Sigma|G, \{\beta_j\}) = \prod_{i \in V} \prod_{c \in C} P_i(\tau_i^c | \Sigma^c, G, \{\beta_j\}), \quad (1)$$

where V is the set of nodes in graph G . Note that we can use our knowledge about the activation times of each node and compute the probability of node i not being activated by its neighbor k (from layer j) before $\tau_i^c - 1$ under cascade c , which is equal to $\prod_{t=0}^{\tau_i^c - 2} (1 - \beta_j \mathbf{1}_{\tau_k^c \leq t})$, where $\mathbf{1}_x$ is the indicator function (it is equal to 1 when x is true). Similarly, the probability of the same node not being activated by its neighbor exactly at time $\tau_i^c - 1$ is equal to $(1 - \beta_j \mathbf{1}_{\tau_k^c \leq \tau_i^c - 1})$. Have in mind that $\tau_i^c = t_{max}$ is equivalent to node i not being activated at all. Then each element of the product in Eq. (1), being the probability of node activation under a specific cascade, can be computed

as follows:

$$P_i(\tau_i^c | \Sigma^c, G, \{\beta_j\}) = \left(\prod_{t=0}^{\tau_i^c-2} \prod_j \prod_{k \in \partial_j i} (1 - \beta_j \mathbf{1}_{\tau_k^c \leq t}) \right) \times \left(1 - \prod_j \prod_{k \in \partial_j i} (1 - \beta_j \mathbf{1}_{\tau_k^c \leq \tau_i^c - 1}) \mathbf{1}_{\tau_i^c < t_{max}} \right), \quad (2)$$

where $\partial_j i$ is the set of neighbors of node i in layer j . If we know one or more layers of the network, we can use the above equation to compute the probability of observed dynamics, assuming that there are no other propagation channels. As seen from Eq. (1), adding more cascades reduces the likelihood. Nevertheless increased statistics of cascades makes it easier to find hidden edges, as it is shown in Fig. 8.

B. Detecting the existence of a hidden layer

An unobserved layer of propagation may lead to a situation where nodes become active despite not interacting, on the observed network, with any other active node. Such a situation leads to the probability described by Eq. (1) being equal to zero. In such a case, one can be sure that there exist spreading paths that are not present in the observed layers. In reality, different layers share certain similarities and may be strongly correlated which would decrease the probability of observing a forbidden activation – in other words, it is less likely that an activation, which occur through a hidden connection, will have a zero probability. Real-life social networks are a good example of this because of a significant overlap between connections on different social media platforms. Moreover, social connectivity is characterised by high clustering [31] which increases the probability of observed neighborhood being active, even though the activation came through an unobserved edge. To address such a situation and still be able to evaluate whether a given set of cascades indicates existence of an unobserved layer or not, we assume a single layer model and utilise the Vysochanskij–Petunin inequality (VP) [32] to evaluate whether a given cascade could be generated by such a model.

We expect that cascades, which involved spreading through unobserved edges, have significantly lower likelihood than those in the assumed null model. Assuming that the likelihood distribution is unimodal, we can quantify how distant the value of an observed cascade likelihood is from the value expected from the single-layer model using the VP inequality. It states that if X is a random variable with a unimodal distribution (mean μ , finite and positive variance σ) and $\lambda > \sqrt{8/3}$, then:

$$\Pr(|X - \mu| \geq \lambda\sigma) \leq \frac{4}{9\lambda^2},$$

which after normalisation $\tilde{X} = \frac{|X - \mu|}{\sigma}$ gives:

$$\Pr(\tilde{X} \geq \lambda) \leq \frac{4}{9\lambda^2}.$$

Inserting $\lambda = \tilde{x}$:

$$\Pr(\tilde{X} \geq \tilde{x}) \leq \frac{4}{9\tilde{x}^2},$$

which means that an upper bound of probability of obtaining a result \tilde{x} or greater from a normalized unimodal distribution describing \tilde{X} is:

$$p(\tilde{x}) = \min\left(\frac{4}{9\tilde{x}^2}, 1\right) \quad (3)$$

for $\tilde{x} > \sqrt{8/3}$. In our case, x is the likelihood of a cascade given by Eq. (1), X is a random variable describing the likelihood values for an assumed single layer model. After performing simulations, we calculate μ and σ , and normalise the x value obtaining \tilde{x} .

In our experience the distributions, which describe X in the single-layer case are unimodal and $\tilde{x} > \sqrt{8/3}$ with a sufficient observation time (t_{max}), thus the assumptions of the VP hold, however, as we are the ones producing these distributions then it is trivial to inspect whether that is the case before proceeding with the rest of our method. In the event, it is not true an alternative theorem, analogous to the VP inequality (e.g. the Chebyshev's inequality [33]), can be introduced or even more general approaches such as bootstrapping [34] can be used to determine the confidence level of the observation.

In the paper, we use $p(\tilde{x})$ to measure how surprising given cascades are, assuming they were generated by an SI process with known β simulated on a visible layer of the network. We validate our approach in the next section, using simulations and synthetic networks.

C. Detecting the hidden edges

In the previous section, we shown how one can discard the possibility that observed cascades were generated by the given network. Once the existence of the hidden layer is established, the same data can be used to estimate the topology of a hidden layer(s). This requires assuming some topology (given visible layer and estimated hidden layer) and β_{hidden} to simulate the process again and calculate new likelihoods. We will try to find the topology by finding cases of node activation that could not be explained by the assumed single layer model. Then for each such activation we identify the set of potential hidden edges. The details of the procedure are as follows:

1. Let $\mathcal{J}^c(t) = \{i \in \mathcal{N} : \tau_i^c \leq t\}$ be a set of nodes that *are infected* in a simulation step t of cascade c . \mathcal{N} represents the set of nodes in graph G and τ_i^c is the activation time of node i in cascade c . $\Delta\mathcal{J}^c(t) = \{i \in \mathcal{N} : \tau_i^c = t\}$ will be a set of nodes that *became infected* in a simulation step t of cascade c .
2. Using above notation we can introduce a set of nodes that became infected in simulation step t of cascade

c but were not a neighbor of any node infected at $t - 1$:

$$\mathcal{U}^c(t) = \Delta \mathcal{J}^c(t) \setminus \bigcup_{m \in \mathcal{J}^c(t-1)} \partial m,$$

where ∂m is the known neighborhood of node m .

3. If the likelihood of a cascade is zero then there is at least one hidden edge in a set:

$$\mathcal{E}(i, c) = \left\{ (i, k) : k \in \mathcal{J}^c(\tau_i^c - 1), i \in \bigcup_{t=1}^{t_{max}} \mathcal{U}^c(t) \right\}.$$

Likelihood of such an edge being the one that was activated to infect i is unknown and difficult to find. However we can say heuristically that said likelihood is:

$$P^c(i, k) \sim (1 - \beta_{hidden})^{|\tau_i^c - \tau_k^c|}.$$

4. Then we must unify the candidates amongst the cascades. Namely if an edge (a, b) was detected in $c = 1$ but not in $c = 2$ we need to associate it with the likelihood of *not being detected* which is similarly non-trivial. Also similarly we can say whatever that likelihood is it must be $\sim (1 - \beta_{hidden})^{|\tau_i^c - \tau_k^c|}$
5. Finally, for each edge e we multiply its likelihoods, therefore obtaining a *joint likelihood* J :

$$J(e) = \prod_c P^c(e).$$

6. Edges that maximise J are most likely the hidden edges we seek.

In order to evaluate the quality of our approach, we will use two metrics:

Sensitivity - the ratio between true-positives and all positives. In our case it is the fraction of hidden edges that were detected. As we simulate our systems many times, we take the mean of the sensitivity across all simulations.

α - Credible Set Size (α -CSS) - a measure introduced in [35]. It represents the number of candidates one must investigate in order to have α level of certainty of finding the sought entity. In practice one computes the rank, i.e., the position, of the entity one wishes to find, in the list of the candidates, which is in a descending order, in accordance to a given measure said entity should maximise. This is repeated many times in order to get a distribution of that rank, and finally, one takes a quantile $q = \alpha$ of that distribution, thus acquiring the α -CSS value. In our case the measure is the *joint likelihood* J described above in point 5., and there are multiple entities – edges – thus we have taken the liberty of adapting said measure such that we take the *highest recorded rank* amongst the hidden edges and follow the rest as usual. I.e., we compute J for the appropriate edges (see the procedure description 1.-6. above), order these edges by their $J(e)$ (highest to

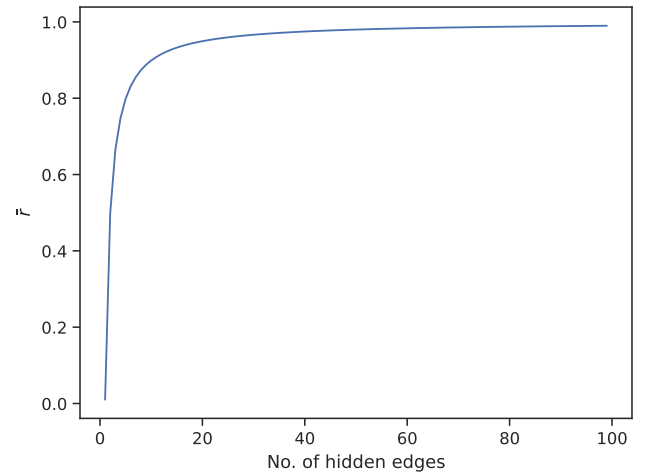


FIG. 2. The ratio of edges required to check in order to have 95% certainty, according to the null model, of testing all hidden edges, as a function of number of hidden edges. The plot is done for a network with $N = 100$ nodes, but the shape of the curve scales with the size of the network.

lowest), find the position of the actual hidden edges (their rank), take the highest recorded rank, repeat 10^4 times, take the quantile $q = 0.95$ of these ranks.

A null model would naturally be random guessing. There are $\binom{N}{2}$ edges to check in a system with N nodes. So for instance, with $N = 100$ that is: $\binom{100}{2} = 4950$ in which case 95% certainty of finding 1 hidden link requires checking $0.95 \times 4950 = 4703$ (rounded up) edges. In general the number r of links required to check in order to have α certainty, according to the null model, can be obtained from:

$$\frac{\binom{r}{k}}{\binom{N}{k}} = \frac{r!}{(r-k)!} \frac{\left(\binom{N}{2} - k\right)!}{\left(\binom{N}{2}\right)!} = \alpha, \quad (4)$$

where k is the number of hidden edges and N is the number of nodes. Fig. 2 presents the normalised $\bar{r} = \frac{2r}{N(N-1)}$ as a function of k in the case of $N = 100$ nodes. As we show later on our method requires substantially less edges to be checked. Since Eq. (4) requires to be solved numerically, we also derive an asymptotic approximation, which can be found in the Appendix B, and from which we can see that $\bar{r} \sim \sqrt[k]{\alpha}$.

III. EXPERIMENTS

We use both real and synthetic data in our experiments. In the latter case we build networks that are realistic and not trivial in the sense that we do not want the occurrence of an activation not explained by the visible network to be likely. To achieve that, we need a way to control the correlation between different layers of the network – where by correlation we mean the percentage of

overlapping edges. Therefore we propose our own models for generating multilayer networks. As for the cascades, these are in both cases generated using the SI model dynamics. As an initial condition for each cascade we randomly pick a node and change its state into *infected*. Each cascade is generate independently.

A. Synthetic Networks

In the first setting, we generate a two-layer network using Barabási-Albert algorithm [36]. There are two parameters, which need to be selected – m_{hidden} and $m_{observed}$, which represent the number of edges added at each step of the algorithm, for hidden and observed layers respectively. The two layers are independent but both have power-law degree distribution, which is believed to resemble real social networks [37] (although lately it is seen more as an idealised approximation [38]). Since lack of correlation makes the problem of detecting hidden layers much easier we also propose another setting. We take a square lattice as the first layer and then apply a rewiring procedure similar to the one introduced by Watts and Strogatz [39] to produce another layer. We start with a square lattice as an equivalent of real relationships (affected by distance) but we also explore scale-free network as a starting layer. The correlation between the two layers is parameterised by p – the probability of each node being rewired to any random (other) node. In all described settings we keep the spreading probability of observed layer equal to $\beta_{observed} = 0.5$. For the hidden layer, this probability takes the values of $\beta_{hidden} = 0.3$ and $\beta_{hidden} = 0.7$.

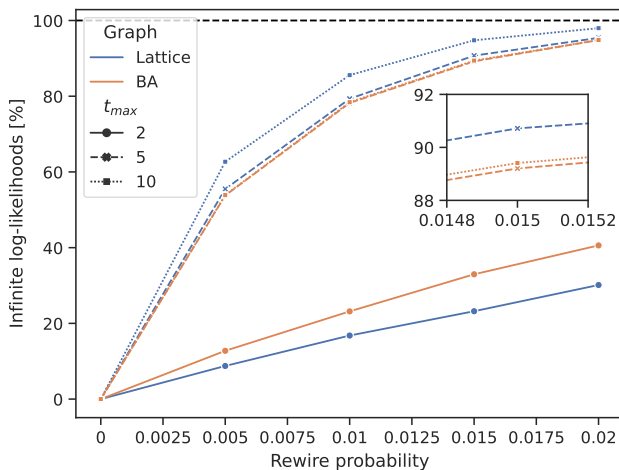


FIG. 3. The percentage of log-likelihoods resulting with $-\infty$ as a function of the probability of rewiring p . We investigate two different cases of networks: a square lattice and the Barabási-Albert network for different lengths of cascades. The simulations were made for networks of size $N = 100$ with periodic boundary conditions in the lattice case. The inset zooms in on the curves obtained for $t_{max} > 2$.

When it comes to detecting hidden layers using any

two-layer network with independent layers results with majority of cascades giving likelihood equal to 0. This is a result of many non-overlapping connections, which activate nodes in a way, which would be impossible for the visible layer. Therefore the problem becomes trivial for such setting. The only interesting case is the model with rewiring as there the correlations between layers can be large. Fig. 3 shows the percentage of cascades resulting with likelihood equal to zero as a function of rewiring probability. The higher the rewiring probability, the more independent are the networks (with rewiring probability of 1 being the fully independent case). We test it for local networks (represented by a square lattice) and when there are many long connections (like in the case of a scale-free Barabási-Albert network). As expected the problem is easier for short cascades and quickly becomes more difficult when the length of cascades grows. Note that even a very small rewiring probability results in a drastic change of the discussed percentage. It practically means that detecting an unknown transmission channel is fairly simple with our approach. A much more challenging task, however, is to find the actual unknown connections. We shall focus on that in further experiments but first let us discuss the case when there is no prohibited dynamics and if we can investigate the likelihood of such observed data.

If the probability of known cascades is positive we can compare it with the empirical distribution of cascades simulated on the observed network. This allows us to use the Vysochanskij–Petunin inequality and decide whether the observed data was generated by process run a graph with an additional (hidden from us) layer. As seen in Fig. 4 using a typical significance level of 0.05 allows to successfully reject the hypothesis about a single layer in significant number of cases (or even all of them). Low t_{max} and β_{hidden} especially in the case of local networks like the square lattice decrease the effectiveness of the test but apart from the extreme case (lattice with $t_{max} = 5$ and $\beta_{hidden} = 0.3$) our proposed approach is an efficient tool for detecting hidden layers.

Once we know that there is a hidden layer affecting dynamics we aim at finding its edges. Tables I and II below show the results of applying our method to both lattice and Barabási-Albert networks with hidden layers produced by rewiring (with probability $p = 0.01$). When comparing the two settings one can observe a certain interplay between sensitivity and α -CSS. For lattice based network the sensitivity is significantly higher than in Barabási-Albert case but at the same time scale-free case is characterised by a much lower α -CSS for both $\alpha = 0.5$ and $\alpha = 0.95$. In other words, it is easier to correctly identify hidden edges when we have a locally connected network (lattice) but at the same time a scale-free network requires a smaller set to find all hidden edges (despite reaching a lower sensitivity level). Note that in both cases the observed 0.5-CSS and 0.95-CSS are significantly lower than for the null model where, depending on the number of rewired links, they would be larger than 2475 and 4703

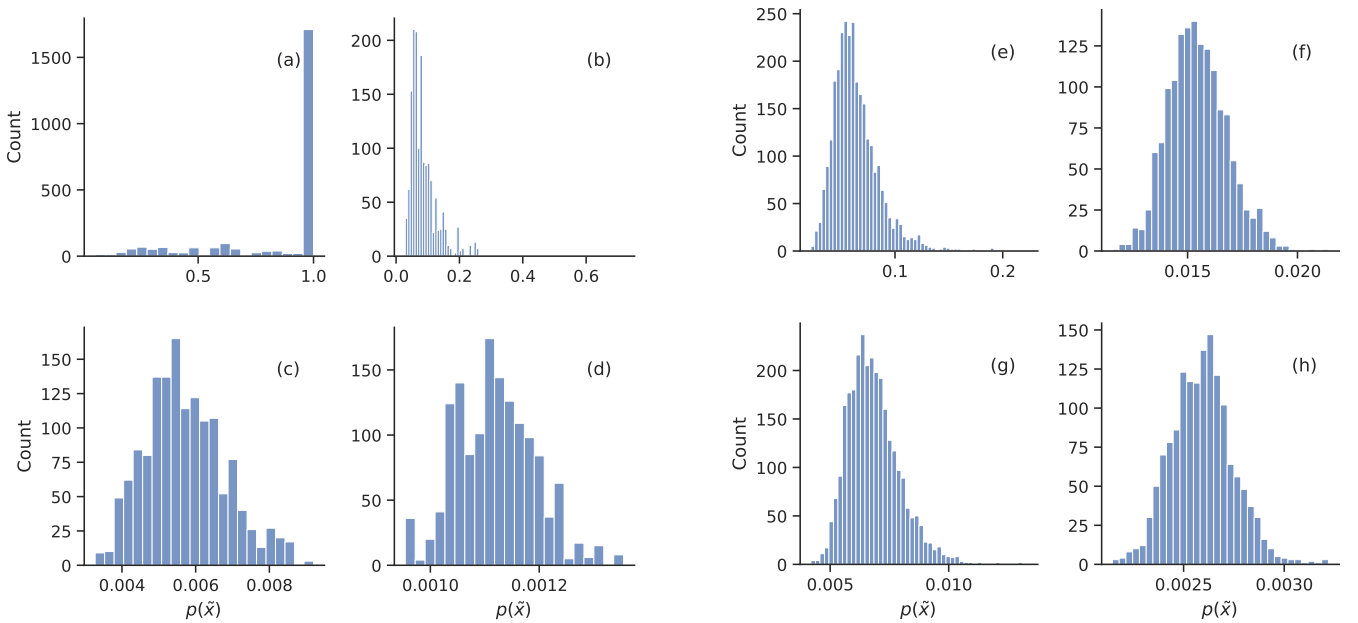


FIG. 4. Histograms of $p(\tilde{x})$ – the probability that the system has only one layer (see Eq. (3)) – for various combinations of t_{max} and β_{hidden} . Plots on the left are generated for square lattice with rewiring, while the ones on the right are generated for BA network with rewiring. Parameters for all the networks are as follows: $\beta_{observed} = 0.5$, $N = 100$, $p = 0.01$ and $m = 3$ (in case of BA networks). Each histogram was made with 10^4 realisations. (a) $t_{max} = 5$, $\beta_{hidden} = 0.3$, (b) $t_{max} = 5$, $\beta_{hidden} = 0.7$, (c) $t_{max} = 10$, $\beta_{hidden} = 0.3$, (d) $t_{max} = 10$, $\beta_{hidden} = 0.7$, (e) $t_{max} = 5$, $\beta_{hidden} = 0.3$, (f) $t_{max} = 5$, $\beta_{hidden} = 0.7$, (g) $t_{max} = 10$, $\beta_{hidden} = 0.3$, (h) $t_{max} = 10$, $\beta_{hidden} = 0.7$,

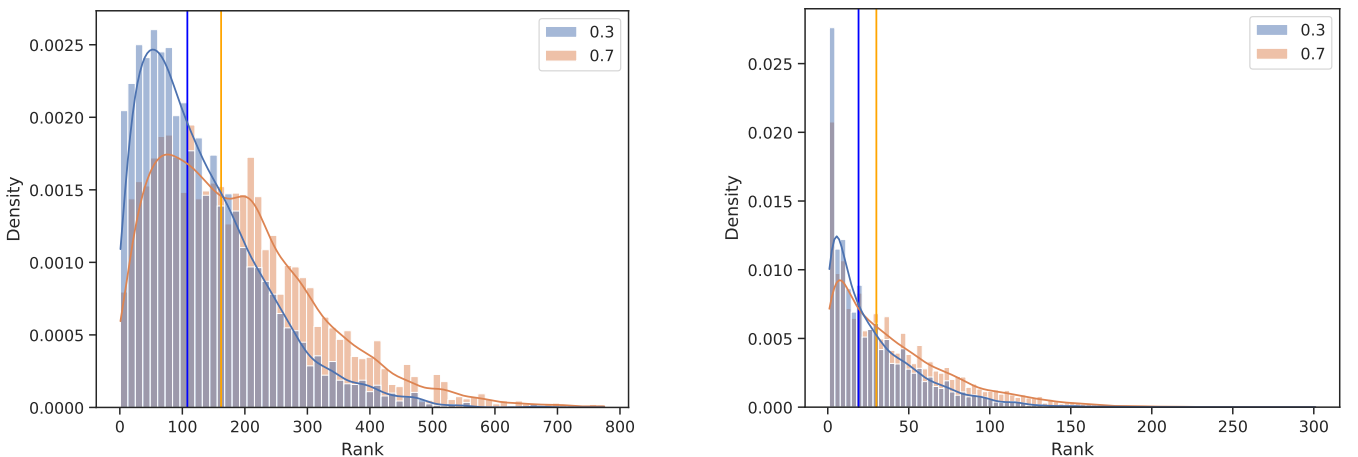


FIG. 5. Distribution of ranks of hidden edges with medians as vertical lines. Left: lattice with rewiring ($N = 100$, $t_{max} = 10$, $p = 0.01$). Right: BA network with rewiring ($N = 100$, $t_{max} = 10$, $m = 3$, $p = 0.01$). Results obtained with 10^4 realisations per scenario where each scenario had 10 independent cascades from different sources. These results are for those realisations where all hidden edges were detected. Solid lines show a Gaussian kernel density estimate.

respectively (see Eq. (4) for $N = 100$ and $k = 1$ – the higher the k , the more links need to be checked for the null-model). Full distribution of ranks from which the α -CSS was computed is shown for both networks at Fig. 5.

As already discussed when the layers are not correlated it is easy to identify that there is a hidden spreading

channel. Nevertheless, finding the actual unobserved links may still be challenging. Both Fig. 6 and 7 show that density of the observed network is an important factor. From the sensitivity perspective it is better to have a denser observed network. Unfortunately the 0.95-CSS also grows with the density of known connections, making it more demanding to find all the connections.

β_{hidden}	$\beta_{observed}$	sensitivity	0.5-CSS	0.95-CSS
0.3	0.5	0.81	108	322
0.7	0.5	0.85	162	422

TABLE I. Sensitivity and α -CSS for a square lattice with rewiring ($N = 100$, $t_{max} = 10$, $p = 0.01$). Results obtained for 10^4 realisations per scenario where each scenario had 10 independent cascades with randomly selected sources.

β_{hidden}	$\beta_{observed}$	sensitivity	0.5-CSS	0.95-CSS
0.3	0.5	0.53	19	87
0.7	0.5	0.69	30	116

TABLE II. Sensitivity and α -CSS for a Barabási-Albert network with rewiring ($N = 100$, $t_{max} = 10$, $m = 3$, $p = 0.01$). Results obtained for 10^4 realisations per scenario where each scenario had 10 independent cascades with randomly selected sources.

Additionally, although the effect is weaker, it is beneficial for both measures if the hidden layer is sparser. This aligns with intuition since more hidden connections can make the observed dynamics much more complex and unsurprisingly having more data about the cascades also makes the task easier. The actual dependence between the number of cascades and the sensitivity is shown in Fig. 8. For a relatively big BA network we need around 30-40 cascades to reach a fairly satisfactory sensitivity level of around 0.8. Depending on the specifics of the problem such amounts of data may be considered a lot (e.g., in epidemic spreading) or easily available (e.g. information spreading on social media). In the next subsection we will see that scaling for real-life networks.

B. Real World Networks

On top of synthetic networks we also use real-world data to build a multilayer network and empirically test our methods. For that purpose we choose the data collected among employees of the Department of Computer Science at Aarhus University [29]. It is a multilayer network consisting of Facebook friendships, co-authorships, work, leisure (repeated leisure activities) and a lunch layer (regularly eating lunch together). Its full structure is presented on Fig. 1 with each layer being shown as a separate network. The whole network has 61 nodes and 620 edges in total.

Main results for the Aarhus data are shown in Tables III, IV and V, which use respectively *facebook*, *work* and *lunch* layers as the visible parts of the graph. We omitted the other two possible cases because of their low density of connections. We treat remaining hidden layers as one, aggregated layer as it does not matter how many layers exactly there are in our detection method. Sensitivity and CSS values are consistent with synthetic results in the sense that they both grow with the density of the visible layer. It is also apparent that our approach far exceeds the

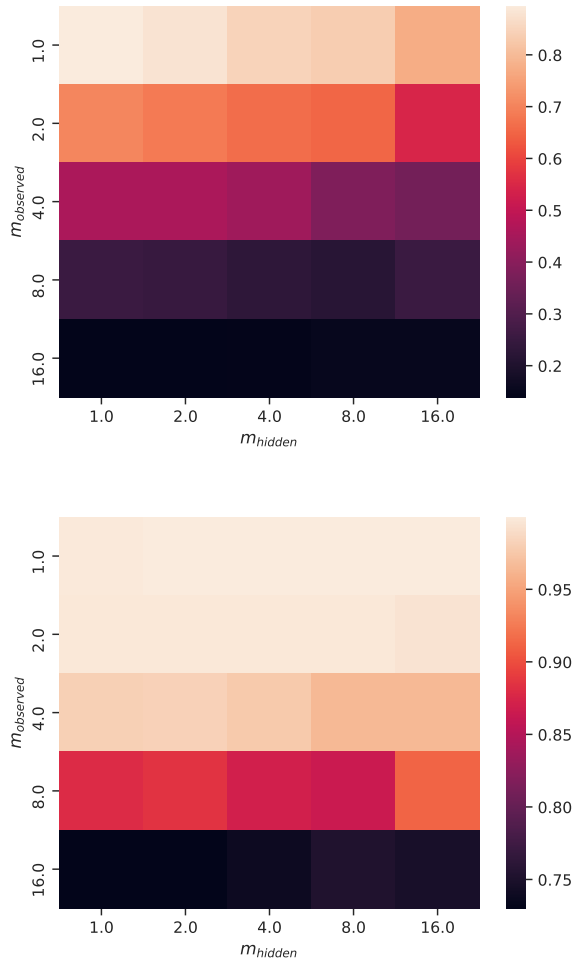


FIG. 6. The sensitivity as a function of m_{hidden} and $m_{observed}$ for two layer Barabási-Albert network with $\beta_{hidden} = 0.7$, $\beta_{observed} = 0.5$ and $t_{max} = 10$ after 10 (top) and 100 (bottom) cascades. The results are averaged over 20 independent runs.

performance of the null model. Independently of which layer will be chosen as the visible one, random guessing would require us to check all possible $\binom{61}{2} = 1830$ links, see Eq. (4) and Appendix B, with $k \in \{159, 160, 229\}$ for *work*, *lunch*, *facebook* as the visible layer respectively. Our method, on the other hand, needs significantly less than that. Here, k is the number of unique edges in the whole graph (353) minus the number of edges in the visible layer. Note that in order to account for the fact that our method does not always find *all* the links, i.e., sensitivity < 1 , one can adjust Eq. (4) such that one multiplies k by the expected sensitivity. That, however, barely changes the result, i.e., null model requires maybe one or two edges less than all possible at best. Moreover, our method gets more successful the more cascades we can observe. A more detailed dependence between sensitivity and the number of cascades is shown in Fig. 8, where different colors represent different observed layers. In Fig. 9 we show

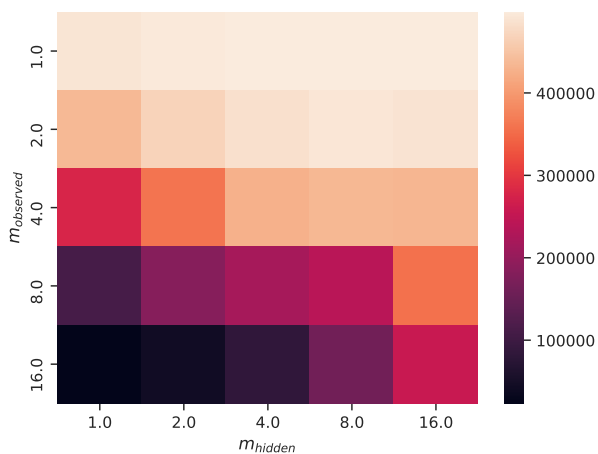
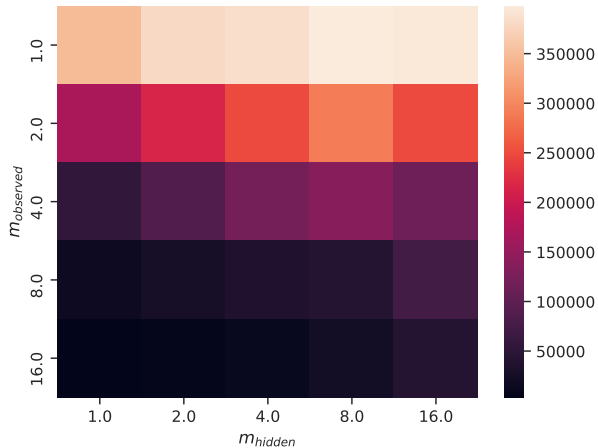


FIG. 7. The 0.95-CSS as a function of m_{hidden} and $m_{observed}$ for two layer Barabási-Albert network $N = 1000$ with $\beta_{hidden} = 0.7$, $\beta_{observed} = 0.5$ and $t_{max} = 10$ after 10 (top) and 100 (bottom) cascades. The results are from 20 independent runs.

the distributions of ranks for the *work* layer as the visible network and when comparing them with the synthetic experiments the two distributions for $\beta_{hidden} = 0.3$ and $\beta_{hidden} = 0.7$ are much more symmetric and separated. The distributions of the other two analysed visible layers are qualitatively similar further supporting the merit of our approach (see Appendix C).

β_{hidden}	$\beta_{observed}$	sensitivity	0.5-CSS	0.95-CSS
0.3	0.5	0.92	1398	1437
0.7	0.5	0.91	1424	1475

TABLE III. Sensitivity and α -CSS for the Aarhus data, with the *facebook* layer as the observed network. Results obtained for 10 cascades with $t_{max} = 10$. Results from 10^4 simulations per β_{hidden} .

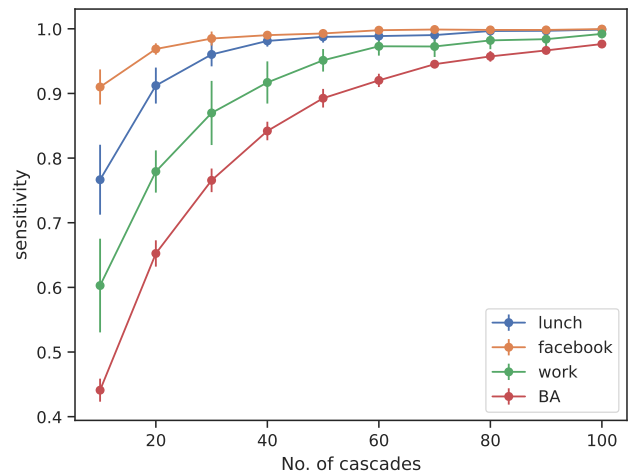


FIG. 8. The sensitivity as a function of number of cascades for a) two layer Barabási-Albert network with $m = 4$ for both layers, $\beta_{hidden} = 0.7$, $\beta_{observed} = 0.5$, $N = 1000$ and $t_{max} = 10$ (red line); b) Aarhus data with different layers as the observed network (*lunch* – blue line, *facebook* – yellow line and *work* – green line). The results are averaged over 20 independent runs and the error bars represent one standard deviation.

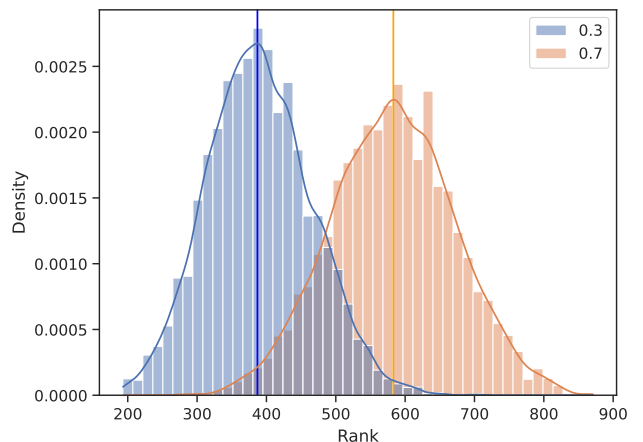


FIG. 9. Distribution of ranks of hidden edges, with medians as vertical lines, for the Aarhus data, with the *work* layer as the observed network. Results obtained for 10 cascades with $t_{max} = 10$, $\beta_{observed} = 0.5$ and two values of β_{hidden} – 0.3 and 0.7. Results from 10^4 simulations per β_{hidden} . Solid lines show a Gaussian kernel density estimate.

β_{hidden}	$\beta_{observed}$	sensitivity	0.5-CSS	0.95-CSS
0.3	0.5	0.42	387	515
0.7	0.5	0.58	583	729

TABLE IV. Sensitivity and α -CSS for the Aarhus data, with the *work* layer as the observed network. Results obtained for 10 cascades with $t_{max} = 10$. Results from 10^4 simulations per β_{hidden} .

β_{hidden}	$\beta_{observed}$	<i>sensitivity</i>	<i>0.5-CSS</i>	<i>0.95-CSS</i>
0.3	0.5	0.71	732	838
0.7	0.5	0.78	841	966

TABLE V. Sensitivity and α -CSS for the Aarhus data, with the *lunch* layer as the observed network. Results obtained for 10 cascades with $t_{max} = 10$. Results from 10^4 simulations per β_{hidden} .

IV. DISCUSSION

Spreading processes on networks are a valuable tool when describing real-life global diffusion processes, like epidemics, information spreading, cascading failures etc. These processes may have several spreading channels and rarely do we know, or are even aware, of all of them. It is therefore crucial to identify whether observed spreading was in fact generated only by the observed network. Furthermore, should one confirm the existence of an unobserved spreading path, finding these hidden connections can be of the utmost importance.

In this paper we focused on identifying both the existence and the structure of a hidden spreading layer by observing a diffusion process unraveling on a graph. We provide methods for i) determining whether a hidden layer exists and ii) estimating what links are present in that layer. Our approach is based on an exact formula for the likelihoods of an observed cascade given knowledge of the system's topology. Using said likelihood and the fact its distribution can be assumed to be unimodal we established a practical and effective way of discerning the existence of a hidden layer. Furthermore using a series of heuristics we obtain an algorithm for estimating the joint likelihood of given (hidden) edge taking part in the observed cascade therefore providing a tool for assessing which nodes are most likely to exchange information via channel we do not know of that is vastly superior to random guessing. In short, a typical situation where our approach could be used is when we observe a single network and a spreading process described by a known model and we suspect that there are hidden layers through which the spread may progress. Note that the observed network can already have multiple layers and the hidden part can also consist of more than one layer. Finally, one could also use our method in a setting where only one layer exist, but at the same time some links are not observed.

Data from synthetic and empirical networks alike confirm that uncovering the hidden spreading channel is a relatively simple task with our approach - especially when the layers are uncorrelated. It is, however, more difficult to identify specific hidden connections. Despite the general similarities there are some quantitative differences between the results obtained with synthetic and real data. One of the most significant differences is how β_{hidden} relates to the distribution of ranks of hidden edges, influencing the difficulty of hidden connections reconstruction.

This effect is much stronger in real world networks than in synthetic ones. It can, however, be explained by the difference in density between hidden and observed networks. In the corresponding plots for synthetic data (see Fig. 5) both layers have the same density. Here the hidden layer is denser (it is a sum of four hidden layers) and so changing the hidden spreading probability affects majority of connections.

An important factor in being able to successfully recover the hidden connections turns out to be the density of both the hidden and the observed transmission layers. Specifically, we observe that the denser the hidden layer the harder it is to find the exact connections. An interesting interplay takes place when it comes to the density of the observed layer. On one hand the sensitivity decreases with the density of observed layer, on the other hand, the α -CSS is also decreasing with the density. This observation, confirmed by both synthetic and real data, means that as the number of connections on a visible layer increases, we are able to identify less hidden edges on average but we need to take into account a smaller set of potential edges in order to find all of the hidden connections.

It should be pointed out that we only focus on the hidden connections which are not overlapping with the observed ones. This means that for correlated layers there might be only few unknown connections whereas the overlapping edges are also influencing the dynamics. Focusing on the more general picture and including the overlapping connections is an interesting subject for future research. Another research direction would be to focus on further improving the hidden connections identification algorithms. These improvements should include both the effectiveness and scalability of proposed methods. The latter is specifically important since real world networks are often quite substantial in size. From the perspective of empirical data it would also be useful to have a way of handling a scenario where different layers have different values of β which may or may not be known. Finally, a more radical generalisations like including temporal networks could also prove to be an interesting research problem. While we do hope to address some of the above topics in the near future we feel that methods presented here already provide effective and practical tools for real world applications.

ACKNOWLEDGMENTS

Ł.G.G and J.C. were supported by National Science Centre, Poland Grant No. 2015/19/B/ST6/02612. M.W. acknowledges support from the Laboratory Directed Research and Development program of Los Alamos National Laboratory under projects numbers 20200121ER and 20210529CR.

-
- [1] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.
- [2] Mikko Kivelä, Alex Arenas, Marc Barthélemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.
- [3] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.
- [4] Manlio De Domenico, Albert Solé-Ribalta, Elisa Omodei, Sergio Gómez, and Alex Arenas. Ranking in interconnected multilayer networks reveals versatile nodes. *Nature communications*, 6(1):1–6, 2015.
- [5] Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.
- [6] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.
- [7] Manlio De Domenico, Clara Granell, Mason A Porter, and Alex Arenas. The physics of spreading processes in multilayer networks. *Nature Physics*, 12(10):901–906, 2016.
- [8] Guilherme Ferraz de Arruda, Francisco A Rodrigues, and Yamir Moreno. Fundamentals of spreading processes in single and multilayer complex networks. *Physics Reports*, 756:1–59, 2018.
- [9] Robert Paluch, Łukasz G Gajewski, K Suhecki, and Janusz A Hołyst. Source location on multilayer networks. *arXiv preprint arXiv:2012.02023*, 2020.
- [10] Sergio Gomez, Albert Diaz-Guilera, Jesus Gomez-Gardenes, Conrad J Perez-Vicente, Yamir Moreno, and Alex Arenas. Diffusion dynamics on multiplex networks. *Physical review letters*, 110(2):028701, 2013.
- [11] Albert Sole-Ribalta, Manlio De Domenico, Nikos E Kouvaris, Albert Diaz-Guilera, Sergio Gomez, and Alex Arenas. Spectral properties of the laplacian of multiplex networks. *Physical Review E*, 88(3):032807, 2013.
- [12] Anna Chmiel and Katarzyna Sznajd-Weron. Phase transitions in the q-voter model with noise on a duplex clique. *Physical Review E*, 92(5):052812, 2015.
- [13] Anna Chmiel, Julian Sienkiewicz, and Katarzyna Sznajd-Weron. Tricriticality in the q-neighbor ising model on a partially duplex clique. *Physical Review E*, 96(6):062137, 2017.
- [14] Anna Chmiel, Julian Sienkiewicz, Agata Fronczak, and Piotr Fronczak. A veritable zoology of successive phase transitions in the asymmetric q-voter model on multiplex networks. *Entropy*, 22(9):1018, 2020.
- [15] Łukasz G Gajewski, Julian Sienkiewicz, and Janusz A Hołyst. Bifurcations and catastrophes in temporal bi-layer model of echo chambers and polarisation. *arXiv preprint arXiv:2101.03430*, 2021.
- [16] Lucas Lacasa, Inés P Mariño, Joaquin Miguez, Vincenzo Nicosia, Édgar Roldán, Ana Lisica, Stephan W Grill, and Jesús Gómez-Gardeñes. Multiplex decomposition of non-markovian dynamics and the hidden layer reconstruction problem. *Physical Review X*, 8(3):031038, 2018.
- [17] Łukasz G Gajewski, Julian Sienkiewicz, and Janusz A Hołyst. Discovering hidden layers in quantum graphs. *arXiv preprint arXiv:2012.01454*, 2020.
- [18] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37, 2012.
- [19] Bruno Abrahao, Flavio Chierichetti, Robert Kleinberg, and Alessandro Panconesi. Trace complexity of network inference. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 491–499, 2013.
- [20] Jean Pouget-Abadie and Thibaut Horel. Inferring graphs from cascades: A sparse recovery framework. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–626, 2015.
- [21] Alfredo Braunstein, Alessandro Ingrosso, and Anna Paola Muntoni. Network reconstruction from infection cascades. *Journal of the Royal Society Interface*, 16(151):20180844, 2019.
- [22] Praneeth Netrapalli and Sujay Sanghavi. Learning the graph of epidemic cascades. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):211–222, 2012.
- [23] Andrey Lokhov. Reconstructing parameters of spreading models from partial observations. In *Advances in Neural Information Processing Systems*, pages 3467–3475, 2016.
- [24] Jiin Woo, Jungseul Ok, and Yung Yi. Iterative learning of graph connectivity from partially-observed cascade samples. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 141–150, 2020.
- [25] Mateusz Wilinski and Andrey Y Lokhov. Scalable learning of independent cascade dynamics from partial observations. *arXiv preprint arXiv:2007.06557*, 2020.
- [26] Vincent Gripon and Michael Rabbat. Reconstructing a graph from path traces. In *2013 IEEE International Symposium on Information Theory*, pages 2488–2492. IEEE, 2013.
- [27] Roger Guimerà and Marta Sales-Pardo. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52):22073–22078, 2009.
- [28] Caterina De Bacco, Eleanor A Power, Daniel B Larremore, and Christopher Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 95(4):042317, 2017.
- [29] Matteo Magnani, Barbora Micenkova, and Luca Rossi. Combinatorial analysis of multiple networks. *arXiv preprint arXiv:1303.4986*, 2013.
- [30] Maureen Hurley, Glen Jacobs, and Melinda Gilbert. The basic si model. *New Directions for Teaching and Learning*, 2006(106):11–22, 2006.
- [31] Harrison C White, Scott A Boorman, and Ronald L Breiger. Social structure from multiple networks. i. block-models of roles and positions. *American journal of sociology*, 81(4):730–780, 1976.
- [32] Friedrich Pukelsheim. The three sigma rule. *The American*

Statistician, 48(2):88–91, 1994.

- [33] John G Saw, Mark CK Yang, and Tse Chin Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.
- [34] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.
- [35] Robert Paluch, Łukasz G Gajewski, Janusz A Hołyst, and Bolesław K Szymanski. Optimizing sensors placement in complex networks for localization of hidden signal source: A review. *Future Generation Computer Systems*, 112:1070–1092, 2020.
- [36] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [37] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.
- [38] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature communications*, 10(1):1–10, 2019.
- [39] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440–442, 1998.

Appendix A: Independent-Cascade model

As mentioned in the main text, our methodology can easily be applied to different spreading processes. One example of such process is the IC model. The factorised form of the likelihood, in case of the IC model, is the same as for the Susceptible-Infected model

$$P(\Sigma|G, \{\beta_j\}) = \prod_{i \in V} \prod_{c \in C} P_i(\tau_i^c | \Sigma^c, G, \{\beta_j\}). \quad (\text{A1})$$

The form of the local probability, however, is different

$$P_i(\tau_i^c | \Sigma^c, G, \{\beta_j\}) = \left(\prod_j \prod_{k \in \partial_{j,i}} (1 - \beta_j \mathbf{1}_{\tau_k^c \leq \tau_i^c - 2}) \right) \times \left(1 - \prod_j \prod_{k \in \partial_{j,i}} (1 - \beta_j \mathbf{1}_{\tau_k^c = \tau_i^c - 1}) \mathbf{1}_{\tau_i^c < t_{max}} \right), \quad (\text{A2})$$

which is only valid for $\tau_i^c > 0$, otherwise it is equal to 1. The plots equivalent to Fig. 3 and Fig. 5 in the case of the IC model are shown respectively in Fig. 11 and Fig. 12. Note that these results do not deviate significantly from the SI simulations. This is interesting for two reasons. First, one could easily produce artificial examples where either IC or SI would be very easier to recover. On one hand it should be easier for IC model to spot any forbidden dynamics, because of the restrictive condition regarding only one time step where spreading is possible for a single node. SI, on the other hand, ensures more statistics, because of the opposite situation (nodes can spread at any time once they are infected). Second, since these are limiting versions of the SIR model, all other variations (where recovery probability is between 0 and $+\infty$) will also produce similar results. This makes

the shown results much more universal, than just for the two analysed models.

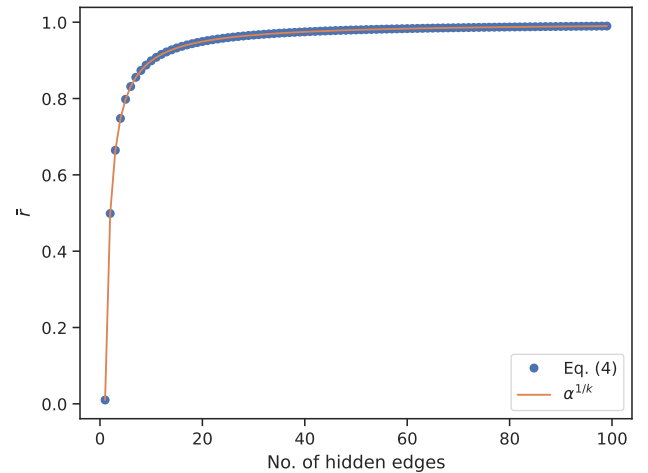


FIG. 10. Comparison of numerical solution of Eq. (4) and the approximation (B3) for $N = 100$. The curve represents the fraction of edges required to check by the null model in order to have 95% certainty of testing all hidden edges.

Appendix B: Null model

In the main text we use Eq. (4) as our null model, however, that form must (for the most part) be solved numerically for r with a set α . Here we present an approximation that gives a closed form for r and provides an excellent match to numerical computations (and is of course much easier and faster to compute).

Let us recall the aforementioned formula:

$$\alpha = \frac{\binom{r}{k}}{\binom{\binom{N}{2}}{k}} = \frac{r!}{(r-k)!} \frac{\left(\binom{N}{2} - k\right)!}{\left(\binom{N}{2}\right)!}. \quad (\text{B1})$$

Denote $\binom{N}{2}$ as ξ , then

$$\begin{aligned} \alpha &= \frac{r!}{(r-k)!} \frac{(\xi-k)!}{\xi!} \\ &= \frac{(r-1)(r-2)\dots(r-k)!}{(r-k)!} \frac{(\xi-k)!}{(\xi-1)(\xi-2)\dots(\xi-k)!} \\ &= \frac{(r-1)(r-2)\dots(r-k+1)}{(\xi-1)(\xi-2)\dots(\xi-k+1)} \sim \left(\frac{r}{\xi}\right)^k. \end{aligned} \quad (\text{B2})$$

We can therefore conclude that

$$r \sim \frac{1}{2} N(N-1) \sqrt[k]{\alpha}. \quad (\text{B3})$$

This leads to:

$$\bar{r} = \frac{2r}{N(N-1)} \sim \sqrt[k]{\alpha}. \quad (\text{B4})$$

The comparison of this result to the numerical solution is shown in Fig. 10.

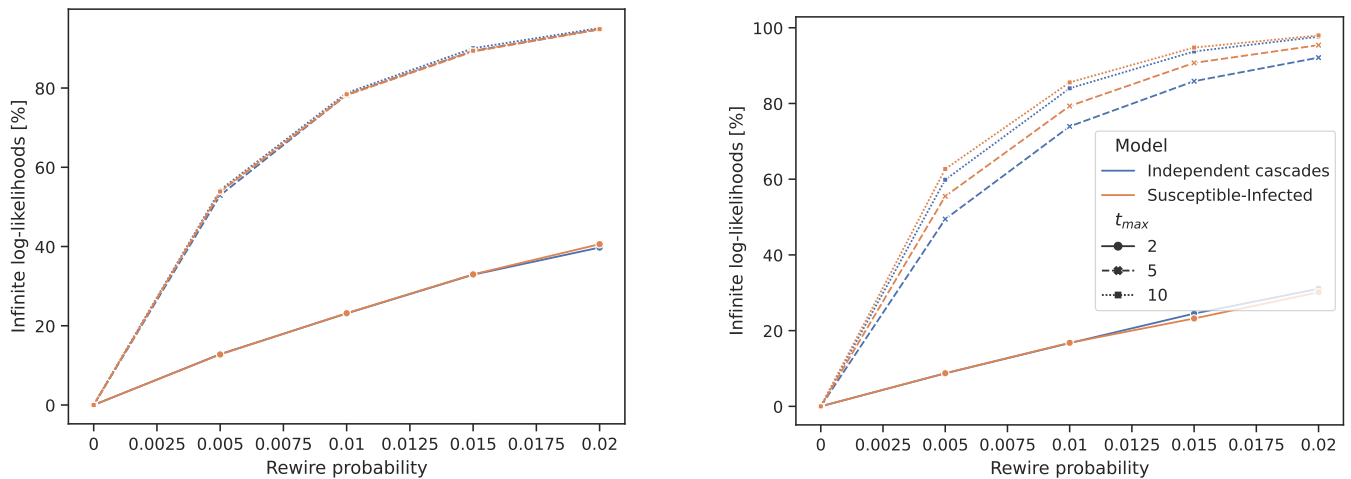


FIG. 11. The percentage of log-likelihoods resulting with $-\infty$ as a function of the probability of rewiring p . Simulations were conducted using two different spreading models - Susceptible Infected (presented in main text) and Independent Cascades for comparison. We investigate two different cases of networks: a square lattice (right) and the the Barabási-Albert network (left) for different lengths of cascades. The simulations were made for networks of size $N = 100$ with periodic boundary conditions in the lattice case.

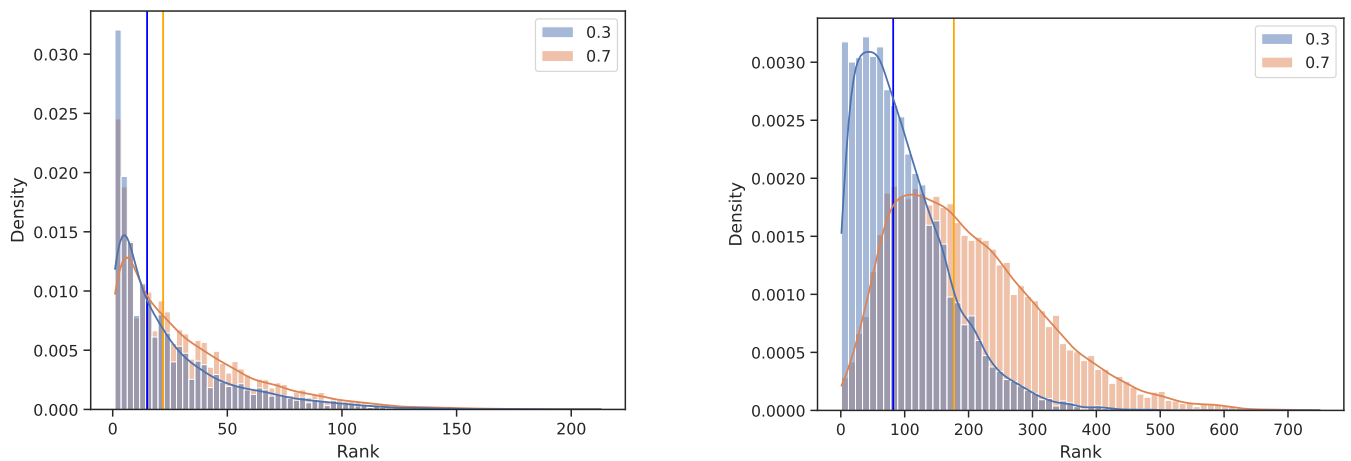


FIG. 12. Distribution of ranks of hidden edges with medians as vertical lines. Simulations were conducted using the Independent Cascade model for comparison against the results from the main text (which show results for the Susceptible Infected model). Right: lattice with rewiring ($N = 100$, $t_{max} = 10$, $p = 0.01$). Left: BA network with rewiring ($N = 100$, $t_{max} = 10$, $m = 3$, $p = 0.01$). Results obtained with 10^4 realisations per scenario where each scenario had 10 independent cascades from different sources. These results are for those realisations where all hidden edges were detected. Solid lines show a Gaussian kernel density estimate.

Appendix C: Distributions of ranks for Aarhus data

We analyse three scenarios for the Aarhus data (described in the main text). For each one of them we use different layer as the visible one and project all the other layers to one hidden network. The distributions of ranks obtained by applying our approach to the *lunch* and *facebook* as observed layer scenarios, are shown in Fig. 13. The third scenario is shown in Fig. 9, in the main text.

Appendix D: Layer detection dependence on β and system size

We test how our layer detection scheme holds for different infection rates and system sizes. By conducting analogous experiments to those presented in Fig. 4 we can show the mean and standard deviation of $p(\hat{x})$ from Eq. (3) as a function of these variables. We set the visible infection rate $\beta_{visible} = 0.5$, $t_{max} = 10$, and simulate 10^4 times per data point.

In Fig. 14 we show results for a BA graph ($m = 3$) and a square lattice with the infection rate of the hidden

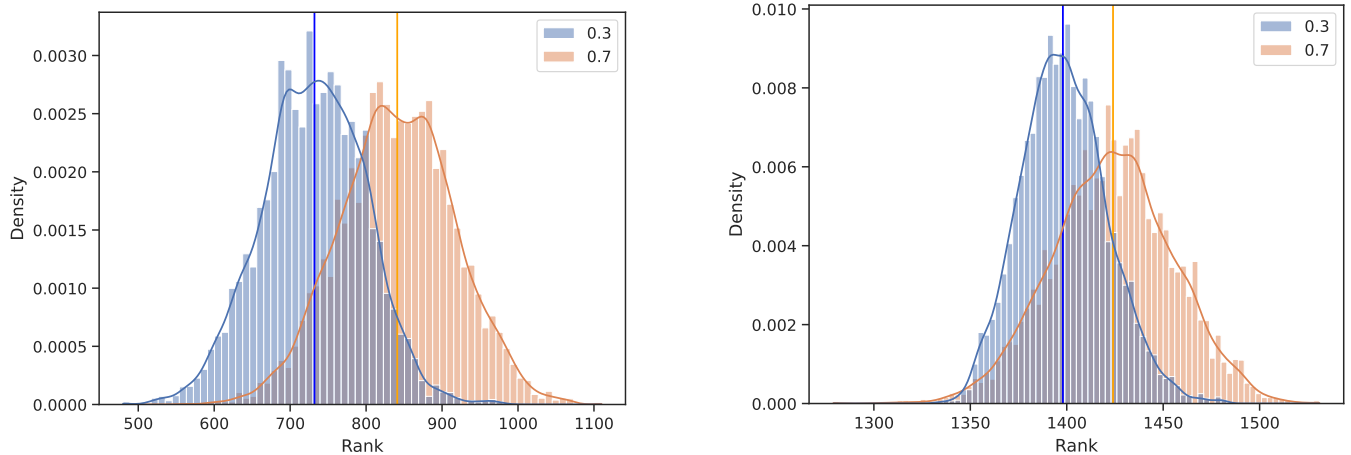


FIG. 13. Distribution of ranks of hidden edges, with medians as vertical lines, for the Aarhus data, with the *lunch* (left) and *facebook* (right) layer as the observed network. Results obtained for 10 cascades with $t_{max} = 10$, $\beta_{observed} = 0.5$ and two values of $\beta_{hidden} = 0.3$ and 0.7 . Results from 10^4 simulations per β_{hidden} . Solid lines show a Gaussian kernel density estimate.

layer $\beta_{hidden} \in [0.1, 1.0]$. When β_{hidden} is small we see a wide spread of the $p(\tilde{x})$ values which is not surprising as the probability of a hidden edge being the preferred connection is also small. As the hidden infection rate rises both the expected value and the variance diminish quickly showing that if the hidden layer is significant enough to cause an effect in the whole system then our method will detect it.

In Fig. 15 we show how the size of the network, $N \in [50, 500]$, affects the $p(\tilde{x})$. On the left we present results for BA ($m = 3$) and on the right for a square lattice. In both cases we choose $\beta_{hidden} = 0.1$ as this is the “worst case” scenario from the previous plot. The blue line shows the mean and one standard deviation of $p(\tilde{x})$ while the

red shows the ratio of infinite log-likelihoods. For the BA graph the statistic is still sufficient in the tested range, however, for the lattice it is not. On the left we can see that as the size increases the mean and variance diminish which is due to the fact that the larger the graph the more potentially “forbidden” links are possible, and thus the detection of the hidden layer becomes easier. On the right the blue curve shows some wild values and that is because of how quickly the red curve rises up to virtually being equal 1. By the time we reach $N = 500$ only an order of 30 realisations out of 10^4 are finite, rendering the blue curve a bit misrepresentative as the statistic is insufficient and biased towards cascades very similar to those generated by a single layer model. We show it here for the sake of completeness.

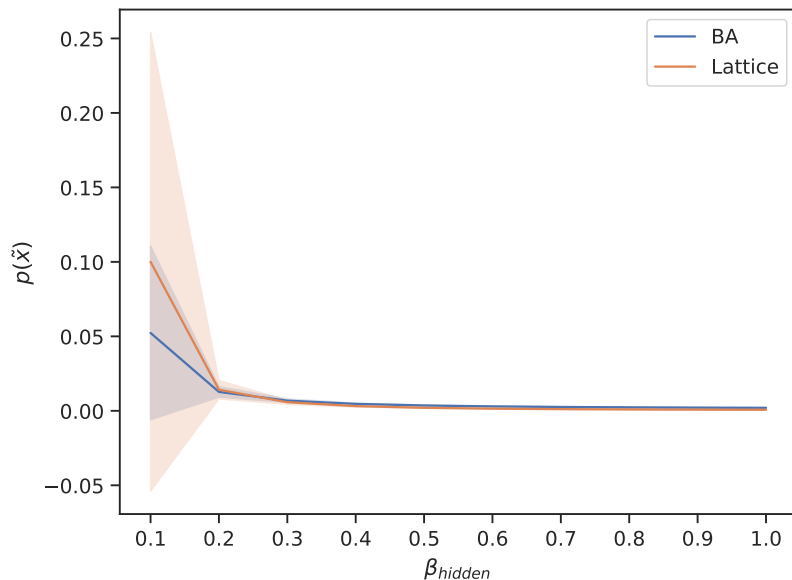


FIG. 14. Mean probability that the system has only one layer as a function of the infection rate on the hidden layer - β_{hidden} - see Eq. 3 and Fig. 4. The colour bands represent one standard deviation. Results for two graphs are presented - BA ($m = 3$) and a square lattice with system size $N = 100$, $\beta_{visible} = 0.5$, and $t_{max} = 10$. Each point is the results of 10^4 simulations.

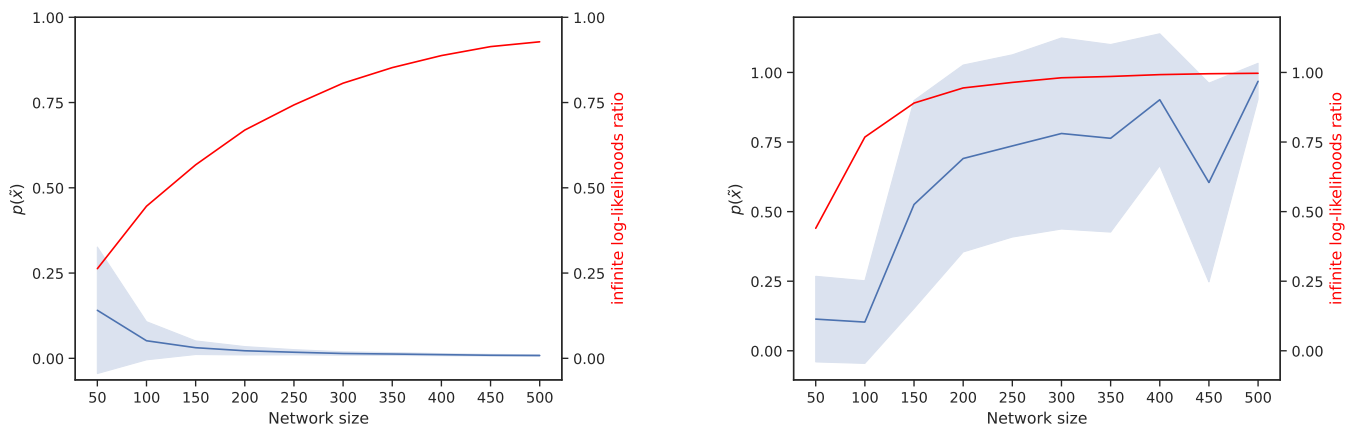


FIG. 15. Mean probability that the system has only one layer as a function of the system size (blue) - see Eq. 3 and Fig. 4. The hidden layer infection rate - $\beta_{hidden} = 0.1$. The colour bands represent one standard deviation. Results for two graphs are presented - BA ($m = 3$, left) and a square lattice (right) $\beta_{visible} = 0.5$, and $t_{max} = 10$. Each point is the results of 10^4 simulations. The red curves (Y-axis values on the right side of the plots) represent the ratio of infinite ($-\infty$) log-likelihoods detected in the data (see Fig. 3). The wide spread and increasing behaviour for the lattice is the result of a very high infinite log-likelihoods ratio - by the time we reach $N = 500$ only an order of 30 realisations out of 10^4 are finite.