In Extended Data Fig. 4b, we extrapolate the infinite size threshold $q_c^{\infty}(\rho)$ to the limit of ∞-body interactions, i.e., for $\rho \to \infty$. The scaling of q_c^{∞} with $1/\rho$ is well consistent with a linear behaviour. We obtain the $\rho = \infty$ limit of $q_c^{\infty}(\rho = \infty) \equiv q_c^{\rm opt}$ from a least-squares fit. For ER networks with average degree $\langle k \rangle = 3.5$ studied here, we find $q_c^{\rm opt} = 0.192(9)$. This is the value of the optimal threshold shown in Fig. 2a in the main text.

V. CI ALGORITHM

We have shown so far that the problem of finding the optimal set of influencers can be solved by minimizing the following cost function which is the leading order approximation in $1/N$:

$$E_\ell(\mathbf{n}) = \sum_{i=1}^{N} z_i \sum_{j \in \partial \mathrm{Ball}(i,\ell)} \left(\prod_{k \in \mathcal{P}_\ell(i,j)} n_k \right) z_j \ , \tag{103}$$

where $E_\ell(\mathbf{n}) = |\mathbf{w}_{(\ell+1)/2}|^2$ for ℓ odd (corresponding to the energy function $\mathcal{E}(\mathbf{n})$ in Eq. S100), and $E_\ell(\mathbf{n}) = \langle \mathbf{w}_{\ell/2} | \hat{\mathcal{M}} | \mathbf{w}_{\ell/2} \rangle$ for ℓ even (corresponding to $\mathcal{E}'(\mathbf{n})$ in Eq. S101). We recall that $z_i = k_i - 1$. We define the collective influence strength at level ℓ, of node i as:

$$\mathrm{CI}_\ell(i) \ = \ z_i \sum_{j \in \partial \mathrm{Ball}(i,\ell)} \left(\prod_{k \in \mathcal{P}_\ell(i,j)} n_k \right) z_j \ , \tag{104}$$

and we can rewrite Eq. (103) as:

$$E_\ell(\mathbf{n}) \ = \ \sum_{i=1}^{N} \mathrm{CI}_\ell(i) \ . \tag{105}$$

Notice that $\mathrm{CI}_\ell(i)$ is basically the same as the fitness b_i of the EO algorithm, and precisely: $\mathrm{CI}_\ell(i) = b_i n_i$. A fast and efficient way to minimize the cost function $E_\ell(\mathbf{n})$ is to adaptively remove the nodes with the highest collective influence $\mathrm{CI}_\ell(i)$. When all the nodes are present, corresponding to $\mathbf{n} = \mathbf{1}$, $\mathrm{CI}_\ell(i)$ evaluates:

$$\mathrm{CI}_\ell(i) \ = \ z_i \sum_{j \in \partial \mathrm{Ball}(i,\ell)} z_j \ . \tag{106}$$

This is the expression of $\mathrm{CI}_\ell(i)$ given in the main text Eq. (5). By computing this quantity for each node, we can find the one with the largest collective influence and then remove it. We stress that the frontier of the Ball: $\partial \mathrm{Ball}(i, \ell)$ consists of all the nodes j that

are at a distance ℓ from i, the distance is measured as the minimum path between i and j. This definition is consistent with the fact that we have neglected the NB walks with loops in the definition of the energy functional Eq. S53 for large networks, and therefore also in $\mathrm{CI}_\ell(i)$ Eq. S106, since they scale as $O(1/N)$ in random networks as discussed in Section II F.

After the removal, the network consists of $N-1$ nodes, and we can proceed as before, looking for the next node with the largest CI_ℓ. Since the removal of the first node changes the degree of its neighbours, we need to decrease their degrees by one before recomputing their CI_ℓ. Removing one by one the nodes according to this adaptive principle we can destroy the network in a nearly optimal and very fast way. Besides, we can significantly speed up the algorithm by decimating a finite fraction of nodes at each step (see Section V B). The algorithm's performance increases by using larger values of the radius ℓ of the Ball(i,ℓ). In Extended Data Fig. 5 we show the results for different values of ℓ. We observe that already for $\ell = 3, 4$ the algorithm reaches the top performance.

When ℓ becomes larger than the network diameter, then $\mathrm{CI}_\ell(i) = 0$. In this situation different nodes are not distinguishable by the algorithm, and thus, the method is basically indistinguishable from a random one. Thus, the parameter ℓ should not exceed in practice the original network diameter. We also notice that dangling ends give zero contribution by CI, and hence they are ignored by the algorithm. This is expected since dangling ends should have zero influence in the network.

The CI algorithm Eq. (5) is based on Eq. (4) which contains the many-body collective interactions that we refer to as "collective influence". The CI algorithm incorporates the collective effects by considering the adaptive nature of the algorithm. The adaptiveness of the CI algorithm, usually called decimation in the spin glass literature [36], is a collective way to select influential nodes, since the removal of each node depends heavily on the history of the process.

A. Optimization for $G(q) \neq 0$

The theory we developed for the optimal fragmentation of networks allows us to compute the optimal influence threshold q_c, i.e. the smallest number of nodes to remove such that $G(q_c) = 0$, together with the corresponding configuration $\mathbf{n}^*$.

When $q < q_c$ the giant component is nonzero, a consequence of the fact that the system of Eqs. (9) has another stable solution different from $\{\nu_{i\to j}\}$ identically zero. Therefore, for $q < q_c$ the stability of the new solution $G(q) \neq 0$ is no more controlled by the non-backtracking operator, but a more complicated operator comes into play that depends on the form of the solution itself. To find the spectrum (or even the largest eigenvalue) of this new matrix we have necessarily to know the solution of the problem. This circumstance depauperates the method of its power in $q < q_c$, since we need the solution to the problem to solve the problem itself. In the regime $q > q_c$, where $G(q) = 0$, this solution can be easily guessed, as we did, but for $q < q_c$ no simple ansatz can be adopted.

What can we do in the regime $q < q_c$ to minimize the size of the giant component?

We know that the configuration $\mathbf{n}^*$ corresponds to a zero giant component. Assuming that this configuration is the optimal one (this hypothesis is not crucial in what follows and can be relaxed by saying that $\mathbf{n}^*$ is the best approximation to the true optimum), then the optimal trajectory in the configuration space, starting from the point $\mathbf{n} = (1, 1, 1, \ldots, 1)$ corresponding to $q = 0$ and $G(0) = 1$, must end up at the point $\mathbf{n}^*$ at $q = q_c$ where $G(q_c) = 0$.

So far we know the final point $\mathbf{n}^*$ and we would like to travel back the optimal trajectory up to the initial point $\mathbf{n} = \mathbf{1}$. In order to do that, let us suppose to decrease infinitesimally the fraction of removed nodes q from its critical value q_c, that is $q = q_c - dq$ [dq can be taken equal to $1/N$, so that $Ndq = O(1)$]. This amounts to explore a neighborhood of the configuration $\mathbf{n}^*$, consisting of a collection of configurations $\mathbf{n}'$ in which a number Ndq of components $n'_i = 0$ is turned into $n'_i = 1$. Here we are making the crucial hypothesis that the optimal trajectory is continuous, in the sense that in going from $q \to q - dq$ only a number Ndq of components n_i is changing state. Under this assumption the trajectory can be followed adiabatically up to the point $\mathbf{n} = \mathbf{1}$.

Mathematically, this can be expressed by saying that the Hamming distance between two neighbouring optimal configurations $\mathbf{x}$ and $\mathbf{y}$: $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} |x_i - y_i|$, is equal to $d(\mathbf{x}, \mathbf{y}) = Ndq$. This hypothesis may not hold in the case where the optimal configuration $\mathbf{x}$ corresponding to Nq removed nodes, and the one $\mathbf{y}$ corresponding to $N(q - dq)$ have an Hamming distance much larger than Ndq. In this case the optimal trajectory has discontinuities, jumping from one point to another which are not close to each other. Physically this correspond to the fact that the optimal state $\mathbf{y}$ cannot be obtained from the optimal state $\mathbf{x}$ by flipping a finite number of components n_i, but requires a global rearrangement of the sys-

tem, which amounts to change the state of much more variables n_i, whose number scales as N^α with exponent $\alpha \in (0, 1]$. This situation takes place in spin-glass systems with Full-RSB thermodynamics, where this chaotic behaviour is observed as a function of the temperature [45] (or as a function of other control parameters like the bond strengths and the magnetic field). In that case, when the system is cooled from a temperature T to $T - dT$ (with T below the critical point: $T < T_c$), the Gibbs state corresponding to the higher temperature does not survive after the cooling, but, instead, a completely new equilibrium state appears at $T - dT$ (i.e. if we sample a typical equilibrium configuration at temperature T, this will be very distant, in the Hamming sense, from a configuration sampled at $T - dT$).

It is highly plausible that the same situation (a chaotic trajectory) is realized also in our problem. To keep things simple, we don't explore this scenario, and analyze only the consequences deriving from the hypothesis of a smooth optimal trajectory from $\mathbf{n}^*$ back to $\mathbf{1}$. This approach will give us a very efficient algorithm to minimize the giant component in all the interval $q \in [0, q_c)$, at no additional computational cost. Therefore we take this performance as a practical justification of the main assumption, leaving to a future work the treatment of the more complicated chaotic scenario.

To take up the threads of our discussion, let us assume that optimal configurations lie close to each other. Knowing the optimal configuration at the fraction q, we should be able to find the new optimal one at $q - dq$ by changing the state of few variables, and actually only one if we take $dq = 1/N$. Practically we have to find the new optimal configuration by changing the state of a single variable from 0 to 1. Practically we proceed in the following way. At $q = q_c$, we have $G(q_c) = 0$. The corresponding configuration n^* contains Nq_c variables $n_i = 0$ (removed nodes), and $N(1 - q_c)$ variables $n_i = 1$ (nodes that are present). At this point we start to add back to the network the nodes using the following algorithm. We assign to each removed node $n_i = 0$ an index $c(i)$, which is calculated as the number of clusters that node i would join if it were reinserted in the network, independently of their sizes. Then we put back in the network node i^* such that $i^* = \text{argmin}_{n_i:n_i=0} c(i)$, by changing $n_i^* = 0$ into $n_i^* = 1$ (see Extended Data Fig. 6). The idea behind this method is the fact that we want to keep a maximally fragmented network after each reinsertion of nodes. We keep on reinserting nodes using the same criterion, until no node is left for which $n_i = 0$. After each reinsertion the indexes $c(i)$ are recalculated, and then the new node with minimum $c(i)$ is chosen.

The running time of this algorithm is $O(MN \log N)$, where M is the number of edges. Indeed, $O(M)$ operations are needed to assign the indexes $c(i)$ and $O(N \log N)$ to sort them. As we did for the case of the main CI algorithm, the time complexity of this algorithm can be reduced to $O(N \log N)$ (if $M = O(N)$), by reinserting a finite fraction of nodes at time.

B. Scalability of the CI algorithm

The time complexity needed to compute the quantity $\mathrm{CI}_\ell(i)$ is proportional to the number of edges inside the ball $\mathrm{Ball}(i, \ell)$. Since the radius ℓ is taken finite, this calculation takes a time $O(1)$ for each node (even if the prefactor increases with ℓ). Thus, to compute the $\mathrm{CI}_\ell(i)$ for all i requires $O(N)$ operations. Sorting the $\mathrm{CI}_\ell(i)$'s takes $O(N \log N)$. The algorithm is terminated when a number Nq_c of nodes is removed. Therefore, removing the nodes one-by-one, the total time complexity would be $O(N^2 \log N)$. Actually we can keep the computational complexity to $O(N \log N)$ without losing any performance, by simply removing a finite fraction of nodes at each step (with a prefactor depending on the percentage of nodes fixed at time). In the next Section we explore the performance of the CI algorithm for different adaptive/decimation steps.

C. Effect of the percentage of fixed nodes during adaptive CI

In this section we show the performance of CI as a function of the percentage of removed nodes at each step of the adaptive algorithm. Indeed, removing a finite fraction of nodes at time reduces the time complexity from $N^2 \log N$ (corresponding to the one-by-one removal) to $N \log N$. In Extended Data Fig. 7 we show the effect of the percentage of fixed nodes at each adaptive step on an ER network of $N = 10^5$ nodes and average degree $\langle k \rangle = 3.5$. As the figure shows, the performance of CI is practically unaffected by the removal of up to 0.25% of nodes at time (i.e. 250 nodes for the considered network) compared to the one-by-one removal.

VI. COMPARISON WITH OTHER HEURISTIC METHODS

In the main text Fig. 2 we compare our solution with heuristics: high-degree and high-degree adaptive [9, 10], PageRank [7], kcore [12], eigenvector [33] and closeness [34] centralities. We also compare in Fig. 3, for Twitter and Mobile Networks, CI with HDA, HD, PR and k-core which are the only heuristics that are scalable to these large-scale datasets. It remains to compare our results with other popular heuristics which do not scale well with system size, and therefore we use smaller systems of 10^4 nodes: betweenness centrality [35] and equal-graph-partitioning [11]. We use the same size and parameters of the scale-free network used in [11]. The final comparison is with BP [14] and it will be done in the next section.

Betweenness centrality (BC) [35]. Betweenness centrality of node i is the sum of the fraction of all-pairs shortest paths that pass through i. BC is a very popular tool for network analysis, which has applications in different fields, from community detection to the human brain. However, it comes with a high computational cost that prevents the examination of large graphs of interest. The best algorithm for BC computations has $O(NM)$ time complexity for unweighted networks with N nodes and M vertices. It is not fast enough, for example, to handle our 10+ million people network. Extended Data Fig. 8 shows its performance. It does not outperform other centralities.

Equal-graph-partitioning (EGP) [11]. This method aims at dividing the network in clusters of equal size. It can behave well for homogeneous networks, like random regular graph, where an equal partition could be expected to destroy the network efficiently, but loses a lot of performance for heterogeneous networks, like scale-free networks, as we can see from Extended Data Fig. 8. Notice that we have used the same network parameters, size, and definition of EGP as given in [11] in the comparison of Extended Data Fig. 8. In fact, we reproduce the same curve and q_c as found for EGP in [11].

VII. COMPARISON WITH BELIEF PROPAGATION ALGORITHM OF ALTARELLI *ET AL.* [14]

The comparison with the Belief Propagation (BP) method proposed in Ref. [14] to optimally immunize a network deserves particular care, because this method does not apply

directly to the problem we are treating here. This is due to the fact that the parameter p in the work of Ref. [14] (which is noted as q in [14]) refers to the fraction of initially infected individuals. In our work the fraction p is assumed to be zero, because in epidemic outbreaks the number of initiators of the epidemic is very small, and typically of order $O(1)$ [46]. For instance, Sierra Leone's explosion of Ebola cases in 2014 appeared to stem from one traditional healer's funeral at which a single source infected 14 women; or the SARS outbreak in 2003 started when one doctor from China infected nine other guests in a Hong Kong hotel who then spread the virus throughout the city and to Vietnam and Canada (source- NY Times August 29, 2014, page A7, "Outbreak in Sierra Leone Is Tied to Single Funeral Where 14 Women Were Infected."). Another example is the patient zero-hypothesis in the AIDS epidemics [47].

On the other hand the model of Ref. [14] is valid for $p > 0$, in particular, the results of Ref. [14] are illustrated for $p = 0.1$. The case $p = 0.1$ would imply an epidemic starting with 10% of the entire population infected independently at the same time. This would imply, for instance, 0.6 million people in Sierra Leone spontaneously and independently being infected at the same time, which would make any targeted immunization intervention perform equally well in practice. This result was shown by Ref. [14] in Fig. 12a: when $p > 0$ any reasonable targeted immunization method gives the same result for the fraction of infected nodes vs immunized nodes. [Fig. 12a in [14] treats the case of $p = 10\%$, noted as $q = 0.1$ in the notation of [14], and compares BP with greedy, HDA, eigenvector centrality and simulating annealing; all showing the same performance].

Therefore, the results of our paper are illustrated for $p = 0$. That being said, next, we compare our results with the BP algorithm in the closest possible regime to ours when $p \to 0$, and also for $p = 0.1$. In the limit $p \to 0$, BP becomes unfeasible because the time complexity of the BP algorithm diverges as p^{-3} for $p \to 0$, as we explain below. The results are shown in Extended Data Figs 9c and 10 and we observe that BP does not perform better than CI. Furthermore, the poor scalability of BP makes it prohibitive for the real networks of 10+ million people used in our work.

To perform a comparison, we need to briefly recall the approach of Ref. [14] and set the notation. The formulation of the problem is based on the long time limit of the SIR dynamics, which is described by the set of variables $\{\nu_i\}$, $i = 1, \ldots, N$, giving the probability for each node to be infected after the epidemic outbreak (in Ref. [14] the variable ν_i is called

m_i, but we prefer to use ν_i to make contact with our notation). These variables satisfy the following equations:

$$\nu_i \;=\; p + (1-p)\left[1 - \prod_{k\in\partial i}(1 - w\nu_{k\to i})\right] \;, \tag{107}$$

where the parameter p is the probability for node i to be initially infected; w is the probability that a given neighbor k of node i transmits the disease to i; and the product on the r.h.s. is over all neighbours k of node i. The variable $\nu_{k\to i}$ (named $m_{k\to i}$ in Ref. [14]) is the probability that node k is infected in a modified network where node i is absent. Each $\nu_{i\to j}$ is associated with a directed edge $i \to j$ of the graph, and satisfies the following equation:

$$\nu_{i\to j} \;=\; p + (1-p)\left[1 - \prod_{k\in\partial i\setminus j}(1 - w\nu_{k\to i})\right] \;. \tag{108}$$

To include the effect of immunization, the authors of Ref. [14] introduce a binary variable σ_i for each node i, taking values $\sigma_i = +1$ if node i is immunized, and $\sigma_i = -1$ if not. Equations (107) and (108) then become:

$$\begin{aligned}
\nu_i \;&=\; \frac{1-\sigma_i}{2}\left\{p + (1-p)\left[1 - \prod_{k\in\partial i}(1 - w\nu_{k\to i})\right]\right\} \equiv\; F_i(\sigma_i, \{\nu_{k\to i}\}_{k\in\partial i}) \;, \\
\nu_{i\to j} \;&=\; \frac{1-\sigma_i}{2}\left\{p + (1-p)\left[1 - \prod_{k\in\partial i\setminus j}(1 - w\nu_{k\to i})\right]\right\} \equiv\; F_{i\to j}(\sigma_i, \{\nu_{k\to i}\}_{k\in\partial i\setminus j}) \;.
\end{aligned} \tag{109}$$

To find the optimal immunization set, Ref. [14] minimizes the following cost (energy) function $E(\sigma, \nu)$:

$$E(\sigma,\nu) \;=\; \sum_{i=1}^{N}\nu_i \;+\; \mu\sum_{i=1}^{N}\frac{1+\sigma_i}{2} \;\equiv\; \sum_{i=1}^{N} e(\sigma_i,\nu_i) \;, \tag{110}$$

where μ is a chemical potential controlling the fraction of immunized nodes. At this point, Ref. [14] applies the cavity method to estimate the single site marginal $P_i(\sigma_i)$, which gives the probability that node i is immunized. Approximating the network with a tree rooted on node i, the authors of Ref. [14] derive the following equation to assess the probability distribution $P_i(\sigma_i)$:

$$P_i(\sigma_i) \;\simeq\; \int\left(\prod_{k\in\partial i} \mathrm{d}\nu_{k\to i}\mathrm{d}\nu_{i\to k} Q_{k\to i}(\nu_{k\to i},\nu_{i\to k})\right) \mathrm{e}^{-\beta e(\sigma_i,\nu_i)} \prod_{k\in\partial i}\delta\left[\nu_{i\to k} - F_{i\to k}\right] \;, \tag{111}$$

where β is the inverse temperature, and the functions $Q_{k\to i}(\nu_{k\to i},\nu_{i\to k})$ satisfy the following

BP equations:

$$Q_{i\to j}(\nu_{i\to j},\nu_{j\to i}) \simeq \sum_{\sigma_i}\int\left(\prod_{k\in\partial i\setminus j}\mathrm{d}\nu_{k\to i}\mathrm{d}\nu_{i\to k}Q_{k\to i}(\nu_{k\to i},\nu_{i\to k})\right)\mathrm{e}^{-\beta e(\sigma_i,\nu_i)}\prod_{k\in\partial i}\delta\left[\nu_{i\to k}-F_{i\to k}\right]\,. \tag{112}$$

Next, we iterate the BP equations to perform a comparison with our approach. These equations do not have an analytical solution, so that, following Ref. [14], we solve them numerically by discretizing the function $Q_{i\to j}(\nu_{i\to j},\nu_{j\to i})$ in a number $\mathcal{N}_{bin}$ of bins. The computational cost to update each message $Q_{i\to j}$ is of order $O(\mathcal{N}^{k_i-1})$, where k_i is the degree of node i. This makes the algorithm practically unfeasible on networks having nodes with large degree (think e.g. to scale free graphs). To overcome this problem, the authors of Ref. [14] use a convolution trick, which reduces the computational cost to $O((k_i-1)\mathcal{N}_{bin}^3)$. Using the convolution method of Ref. [14], Eq. (112) reads:

$$\begin{aligned}Q_{i\to j}(\nu_{i\to j},\nu_{j\to i}) \;\simeq\; \mathrm{e}^{-\beta\mu}\left[\prod_{k\in\partial i\setminus j}\int\mathrm{d}\nu_{k\to i}\;Q_{k\to i}(\nu_{k\to i},0)\right]\;\delta(\nu_{i\to j})\;+\\ +\;\frac{1}{1-p}\;M^{(k_i-1)}\left(\frac{1-\nu_{i\to j}}{1-p},\;(1-w\nu_{j\to i})\frac{1-\nu_{i\to j}}{1-p}\right)\;\mathrm{e}^{-\beta[1-(1-\nu_{i\to j})(1-w\nu_{j\to i})]}\;\Theta(\nu_{i\to j}-p)\;,\end{aligned} \tag{113}$$

where the function $M^{(n)}(x,y)$ is defined iteratively by the convolution:

$$\begin{aligned}M^{(n)}(x,y) &= \int_0^1\mathrm{d}x_1\mathrm{d}x_2\;\delta(x-x_1x_2)\;M^{(n-1)}(x_1,y)\;M^{(1)}(x_2,y)\;,\\ M^{(1)}(x,y) &= \int_0^1\mathrm{d}\nu\;\delta[x-(1-w\nu)]\;Q\left(\nu,1-(1-p)\frac{y}{1-w\nu}\right)\;.\end{aligned} \tag{114}$$

In all the following numerical results we will always use the efficient form (113) of the BP equations.

From the knowledge of the functions $Q_{i\to j}(\nu_{i\to j},\nu_{j\to i})$, Ref. [14] computes the probability distribution $\mathcal{Q}_i(\nu_i)$ that node i has been infected during the epidemics [ν_i is defined in the first line of Eq. (109)], which is given by:

$$\mathcal{Q}_i(\nu_i)\;\simeq\;\mathrm{e}^{-\beta\mu}\left[\prod_{k\in\partial i}\int\mathrm{d}\nu_{k\to i}\;Q_{k\to i}(\nu_{k\to i},0)\right]\;\delta(\nu_i)+\frac{\mathrm{e}^{-\beta\nu_i}}{1-p}\;M^{(k_i)}\left(\frac{1-\nu_i}{1-p},\;\frac{1-\nu_i}{1-p}\right)\;\Theta(\nu_i-p)\,. \tag{115}$$

Moreover, they estimate the single spin marginal $P_i(\sigma_i)$ as:

$$P_i(\sigma_i)\;\simeq\;\mathrm{e}^{-\beta\mu}\left[\prod_{k\in\partial i}\int\mathrm{d}\nu_{k\to i}\;Q_{k\to i}(\nu_{k\to i},0)\right]\;\delta(\sigma-1)+\left[\int\mathrm{d}x\;\mathrm{e}^{-\beta[1-(1-p)x]}\;M^{(k_i)}(x,x)\right]\;\delta(\sigma+1)\,. \tag{116}$$

Once the authors of Ref. [14] obtained the probability distributions $\mathcal{Q}_i(\nu_i)$ and $P_i(\sigma_i)$, they can compute the average fraction of infected nodes f:

$$f = \frac{1}{N}\sum_{i=1}^{N} \int \mathrm{d}\nu_i \; \nu_i \; \mathcal{Q}_i(\nu_i) \; , \tag{117}$$

and the average fraction of immunized nodes q:

$$q = \frac{1}{N}\sum_{i=1}^{N} \frac{1+\langle\sigma_i\rangle}{2} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left(1+\sum_{\sigma_i} \sigma_i \; P_i(\sigma_i)\right) \; . \tag{118}$$

A. BP adaptive

Before we compare BP with our method, we need to illustrate the BP method on a ER network to clarify some technical issues. We consider a small ER random graph of $N = 200$ nodes, where BP can be studied, and average degree $\langle k\rangle = 3.5$. We use the following values of the parameters: fraction of initially infected nodes $p = 0.1$, inverse temperature $\beta = 3.0$, and transmission probabilities $w = 0.4, 0.5, 0.6, 0.7$. The results are shown in Extended Data Fig. 9a, where we plot the fraction of infected nodes f versus the fraction of immunized nodes q. As already noticed in Ref. [14], we observe that, while for $w = 0.4$ the curve is continuous in the whole range of values of q, for larger values of w the curves get interrupted at a certain value of q. This is due to the fact (as mentioned by the authors of Ref. [14]) that the free-energy is non-convex in that region of values of q, and the chemical potential is flat as shown in Extended Data Fig. 9b. Therefore, all values of q in that region cannot be explored using the normal BP method. Physically, the fact that the thermodynamical potential becomes non-convex is the signature of a phase transition happening at a certain value of w. To overcome this problem, the authors of Ref. [14] suggest the following technique. One adds an extra magnetic field H to the energy function $E(\sigma, \nu)$ in Eq. (110), which is then adjusted at each update of the BP equations to keep fixed the value of immunized nodes q. We implemented this adaptive BP method (called 'fixed density BP' in Extended Data Fig. 9a), and we found that the missing part of the curve can be effectively reconstructed for some values of w larger than $w = 0.4$. Nonetheless, for even bigger values of w, we found that the missing part of the curve cannot be fully reconstructed, since the adaptive algorithm does not converge anymore. Usually the non-convergence of the BP algorithm is associated to the existence of a phase transition (different from the

aforementioned one), marking the limit of validity of the replica-symmetric cavity method. We then expect that for those values of q, where reconstruction is impossible, a different BP method has to be used, in order to deal with the phenomenon of replica symmetry breaking.

In the next Section we compare BP with CI in two different settings. The first case is the closest one to the regime where CI is defined (i.e. when $p \to 0$ and $w \to 1$). The second type of comparison is the case where $p > 0$ and the BP method can be used for all values of $q \in [0, 1]$.

B. Comparison

1. *First comparison*

Here, we compare BP in the closest possible regime to CI, i.e. for $p \to 0$ and $w \to 1$. Solving numerically the BP equations requires to discretize the functions $Q_{i\to j}(\nu_{i\to j}, \nu_{j\to i})$ in a number $\mathcal{N}_{bin}$ of bins of the order $\mathcal{N}_{bin} \sim 1/p$, in order to have good numerical accuracy, because the smallest possible non-zero value assumed by ν_i is $\nu_i = p$, as stated in Ref. [14]. The BP running time is of order $O(M\mathcal{N}_{bin}^3)$, M being the number of edges in the graph. The factor $\mathcal{N}_{bin}^3$ comes from the computation of the function $M^{(n)}(x, y)$ in Eq. (114), that requires a double integration over x_1 and x_2 (giving a factor $\mathcal{N}_{bin}^2$), for each value of y (giving an extra factor $\mathcal{N}_{bin}$). Since $\mathcal{N}_{bin} \sim 1/p$, the BP running time diverges as p^{-3} for $p \to 0$. This is the reason why we cannot use BP directly for $p = 0$. So, we set $p = 0.01$, as small as possible, and $w = 0.99$, as close to 1 as possible, in the BP algorithm. Moreover, we choose a quite high value of the inverse temperature $\beta = 10$, close enough to the zero temperature limit. Note that for this value of $p = 10^{-2}$, the number of bins needed for good numerical resolution is of the order of $\mathcal{N}_{bin} \sim 10^2$, which introduces a prefactor in the computational cost of the algorithm already of order 10^6.

We compare CI and BP on a small ER network of $N = 10^3$ nodes and average degree $\langle k \rangle = 3.5$, where BP can be run efficiently to do a study over the parameter space. Since for those values of p and w we cannot compute the full curve $f(q)$ (for the reasons explained in Section VII A), we compare the giant component found by BP and the one obtained with CI (notice that f coincides with the giant component G in the limits $p \to 0$ and $w \to 1$). In order to choose which nodes have to be removed according to BP, we use the following

criterion: we run BP and we assign to each node the value of the sign of its magnetization: $\mathrm{sign}(\langle\sigma_i\rangle)$ (the value of the inverse temperature we chose, $\beta = 10$, is sufficiently high for the spins σ_i to be highly polarized). Then, node i is removed if $\mathrm{sign}(\langle\sigma_i\rangle) = 1$, and it is not if $\mathrm{sign}(\langle\sigma_i\rangle) = -1$. When BP does not converge, we stop the algorithm after a maximum number of iterations and we use the unconverged marginals to assign the magnetizations. In this way we can draw the full curve $G(q)$ even if BP does not converge. The result of the comparison is shown in Extended Data Fig. 9c, where we can see that BP is not better than CI, and performs slightly worst than the HDA method.

To conclude this section, we mention two other versions of the BP algorithm. The first one is developed in Ref. [48]. The technique used in Ref. [48] is the same BP technique as the one introduced by Altarelli *et al.* [13, 14]. From the analytical point of view, Ref. [48] improves the lower bound on the threshold q_c by considering the effects of 1 step replica symmetry breaking (1RSB), obtaining slightly larger lower bounds than those predicted by the replica symmetry (RS) approach of Altarelli *et al.*: $q_c^{\mathrm{RS}} \leq q_c^{\mathrm{1RSB}}$, or $\theta_{\mathrm{min},0} \leq \theta_{\mathrm{min},1}$, respectively in the notation of Ref. [48]. Hence the lower bound in Ref. [48] is larger than the one obtained by Altarelli *et al.*

The second variant of the BP algorithm is used in Refs. [49, 50] for solving the undirected feedback vertex set problem. This algorithm, named Belief Propagation Guided Decimation (BPD), improves the time complexity of the BP approach of [13, 14, 48] and can be tested in SF networks. In Extended Data Fig. 9d we compare the BPD with CI algorithm where we find evidence of the best performance of CI.

2. Second comparison

In this section we compare CI and BP in a different way; this time in the case where BP is well defined and CI is not, for parameter values $p \neq 0$ and $w \neq 1$. Thus, we use $p = 0.1$ and $w = 0.5$. So, this second comparison represents the opposite situation with respect to the previous one. We compare the two methods in the following way. We use BP to compute directly the fraction of infected nodes $f(q)$ as a function of the fraction q of immunized nodes by means of Eqs. (117)–(118). To compare against CI, we have to simulate explicitly the SIR process, since we cannot estimate directly the $f(q)$. More precisely, we first identify the immunized nodes with CI, and then we run the SIR algorithm to obtain the final fraction

of infected individuals $f(q)$.

The result of the comparison is reported in Extended Data Fig. 10, for an ER network of $N = 10^3$ nodes and average degree $\langle k \rangle = 3.5$. The values of the initially infected individuals p and the transmission probability w are $p = 0.1$ and $w = 0.5$. The value of the inverse temperature β used in the BP algorithm is $\beta = 10$ (for the portion of the curve where the adaptive BP algorithm is needed, we chose the lowest possible temperature such that the algorithm converges). As the figure shows, there is little difference between BP and CI, with CI slightly better for small q. Moreover we checked that even using HDA gives more or less the same results as BP and CI, as the authors of Ref. [14] also show in Fig. 12a of their work. Therefore, in the case when $p > 0$ (meaning that a finite fraction of the entire network is already infected from the very beginning of the epidemic outbreak), any reasonable targeted immunization technique gives the same result. That is, the optimization achieved by any method is washed out by the large number of already infected people, and all strategies perform equally well. On the contrary, in the case when $p = 0$, i.e. when the epidemic is initiated by a superspreader event $O(1)$, different strategies behave very differently, with CI being the best so far.

We notice, en passant, that the analytical BP estimation of $f(q)$ gives a lower bound on the actual $f(q)$. That is, if we used the same procedure as for CI, by first identifying the immunized nodes and then computing the fraction of infected ones trough the outcome of the SIR process, the resulting curve $f(q)$ would lie above the analytical BP estimation.

Finally, we note that EO estimates the optimal numerical value of the threshold q_c as a numerical extrapolation to $N \to \infty$ and $\ell \to \infty$. While EO can estimate this threshold accurately (providing an upper bound very close to the real optimum), it cannot provide the actual optimal configuration $\mathbf{n}^*$ for large system sizes. This is of course a general feature due to the NP-hardness of the problem.

Indeed, the EO method we use to estimate the value of optimal threshold for ER random graphs in Extended Data Fig. 4b may not be the only way to assess analytically that result. Indeed, there are other methods to approximate the location of the optimal threshold, which can provide lower or upper bounds. For instance, the BP (or cavity) method investigated above writes down approximate self-consistent equations for the optimization problem, that are solved iteratively to get an estimation of the optimal threshold. Often, the BP equations do not converge (as a consequence of the NP-hardness of the problem), but an attitude has

gained a foothold in the statistical physics community, which amounts to ignore convergence problems and use anyway an unconverged solution as an estimation of the optimal threshold. Indeed, in all cases where this approach has been pursued, it has been shown that the BP analytical prediction provides a lower bound to the optimal threshold. On the contrary, the EO method employed in our work provides an upper bound to the optimal threshold. Therefore, different analytical methods can give, indeed, predictions which are close to each other and, hence, close to the optimal value of the threshold.

Furthermore, it may not be impossible to find the exact analytical value of the threshold even if the problem is NP, as in the case of the Sherrington-Kirkpatrick model for spin glasses [36], where the Parisi ansatz provides the correct solution. We also notice that analytical solutions are based on the analysis of the most probable case in general, but not for a specific instance of the problem. Indeed, not every NP-complete problem can be analysed in this way. Some problems do not permit a discussion based on the most probable case. A random chosen satisfiability problem, for example, is almost always easy to solve, because a random sequence of symbols almost always does not make sense.

In our problem of optimal percolation, even though the numerical value of the threshold could be known exactly with EO or other method, the main problem remains open: finding an optimal configuration that is as close to the minimal as possible in the large system size. The most relevant challenge for practical applications of NP problems is not to estimate theoretically the value of the threshold q_c, but to find a scalable algorithm (for realistic applications should be at most $O(N \log N)$) which is able to approximate as close as possible a real optimal configuration $\mathbf{n}^*$ at q_c.

Our algorithmic solution to this NP problem is then CI: a scalable algorithm $\sim O(N \log N)$ that contains the physics of the optimal configuration, and it is necessarily an approximation to the true optimum; being a $O(N \log N)$ algorithm it cannot give the optimal solution unless P = NP. Thus, proper benchmarking does not compare the analytical value of the threshold q_c. Benchmarking should be carried out by comparing the optimal configurations with the corresponding giant components for large size networks, which should be at least of the order of 10^7+ nodes (as we have done in Fig 3d), showing an improvement both in the running time and efficiency.

VIII. A NEW PARADIGM OF INFLUENCE IN SOCIAL MEDIA: TWITTER

In the next two sections we show that the performance of our method is confirmed in two real networks. We study two prototypical examples of real networks: Twitter web and a social network derived from phone calls. The former is used to test our theory as a new paradigm of influence, while the second can be used to design an immunization protocol in the case of an epidemic outbreak.

We have paved the way to explore the consequences of our theory in real networks, where the assumption of tree-like structure that is the basis of our theory is not necessarily satisfied. The reason to be interested in such a kind of problem is that it is manifestly in the interest of man's communal existence to understand how people increase their influence when they tie one another. The critical question is to what extent one can define a measure for influence solely on the basis of social contact network. The answer might be hard to find, but, at the same time, one cannot deny that the network itself mirrors the mutual relations of users, and hence it must contain information about their influence. The resulting network-based influence estimation can always be supplemented by measures of activity and engagement.

With this caveat in mind, our optimal percolation theory uncovers the optimal influencers in social media. In this context, the measure of node-influence in social media is the drop in the size of the giant cluster which would happen if the node in question were removed. Such a measure of influence is related to the ability to spread the news to the largest portion of the network as shown by our mapping of the maximal spreading problem in LTM (with $\theta_i = k_i - 1$) to optimal percolation. We test this idea in Twitter, next.

Twitter is the online social networking and microblogging service that has gained worldwide popularity. Here we use the dataset of approximately 16 million tweets sampled between January 23rd and February 8th, 2011 and publically shared by Twitter (`http://trec.nist.gov/data/tweets/`) (also available at `http://jamlab.org`, see Ref. [19] for more details). The natural way to get the social network is to extract the follower network through Twitter API. Unfortunately, due to the access rate limit of Twitter API, it is impossible to obtain the full information of the follower network in a reasonable time. Furthermore, many of the follower links are not active. To approximate the social network, we use an alternative way - the mention network [19]. In contrast to the normal tweets, mentions are tweets containing @username and usually include personal conversations or

references. In fact, the mention links have stronger strength of ties than follower links. Therefore, the mention network can be viewed as a stronger version of interactions between Twitter users. In the mention network, if user i mentions user j in his/her tweets, there exists a link from i to j. In order to better represent the social contacts, we also add to the network the retweet relations from the tweets. A retweet (RT @username) corresponds to content forward with the specified user as the nominal source. If user i retweets a tweet of user j, then a contact is established between j and i. We then consider all links to be undirected. In this way, the social network of Twitter is constructed. The resulting network has $N = 469,013$ nodes and $M = 913,457$ edges.

We measure the collective influence of a group of nodes as the drop in the size of the giant component which would happen if the nodes in question were removed. The results are shown in Fig. 3a, showing the better performance of CI in comparison with HDA, PR, HD and k-core. The other heuristics and BP cannot be run in this large dataset.

In Fig. 3b we plot the percentage of fake influencers (PFI) or false positives as a function of the fraction of removed nodes q. This quantity is defined with respect to the HD method, and represents the amount of different influencers between HD and CI. More precisely, we call $S_{\rm CI}(q)$ the set of influencers (i.e. removed nodes) found by CI at a given value of q:

$$S_{\rm CI}(q) = \{x^1_{\rm CI}, x^2_{\rm CI}, \ldots, x^{Nq}_{\rm CI}\} \ , \tag{119}$$

and $S_{\rm HD}(q)$ the corresponding vector for HD. Moreover we denote by $|S(q)| = Nq$ the size of the set. Notice that $|S_{\rm CI}(q)|$ is upper bounded by $Nq_c^{\rm CI}$, i.e. $|S_{\rm CI}(q)| \leq Nq_c^{\rm CI}$, where $q_c^{\rm CI}$ is the influence threshold obtained with CI. Indeed $Nq_c^{\rm CI}$ is the maximum number of influencers. Analogously, $|S_{\rm HD}(q)|$ is upper bounded by $Nq_c^{\rm HD}$.

We define PFI(q) as:

$$\mathrm{PFI}(q) = 100 \times \left[1 - \frac{|S_{\rm CI}(q) \cap S_{\rm HD}(q)|}{Nq}\right] \ . \tag{120}$$

In other words, we measure the percentage of different nodes removed by CI and HD. As shown in Fig. 3b, the PFI at the critical threshold of CI is $\mathrm{PFI}(q_c^{\rm CI}) \sim 26\%$, meaning that HD misses roughly 1/4 of the total number of (real, i.e. optimal) influencers. As a consequence the giant component for HD is still very large, $G_{\rm HD}(q_c^{\rm CI}) \sim 0.37$, and hence, HD needs to keep on removing nodes to fragment completely the network. This comes at the price of including a large number of fake influencers at the end of the process $q_c^{\rm HD}$, where $\mathrm{PFI}(q_c^{\rm HD}) \sim 48\%$.

In the same way, if one knew the true (optimal) configuration of influencer, one could analogously define the fake CI influencers which do not overlap with the optimal ones. Actually, the obtained nearly optimal set by the CI method has an unknown overlap with the true optimal solution. On the other hand, the impossibility to find such an optimal set (because of the aforementioned prohibitive running time), makes the CI influencer set the natural substitute for the optimal one, and, hence, the reference set for studying the overlap with node sets identified by other methods.

IX. HALTING EPIDEMICS: MOBILE PHONE CALL NETWORK

While medicine has made solid advances in the isolation of new vaccines for an increasing number of diseases, and may expect to make still greater ones, no certain claim can be established for a corresponding advance in preventive immunization.

It is of deep social importance to have a fast and optimal intervention strategies when new outbreaks of disease break out. Prevention methods are still limited for various reasons: many virus are responsible of diseases of animals that can be transmitted to humans and thus causing epidemics. It is difficult, if not impossible, to control the populations of vectors and natural reservoirs, or predict what changes in the environment can favor the epidemics. The development of new drugs is usually not the solution to the problem.

It is generally accepted that an efficient way to fight epidemic diseases is the execution of immunization protocols and fast quarantine procedures, together with the spread of the knowledge of these dangers and the efforts to remove the environmental causes that favor them [46]. Probably a certain percentage of diseases will always remain undefeated, but if only one can succeed in reducing to a minority the majority that is today vulnerable, one will have accomplished a great deal, perhaps indeed everything that can be accomplished. In this situation is highly desirable to have a guiding strategy which enables to select who must be vaccinated or put in quarantine. Our theory offers a protocol of selection, the closest one to the optimal. This result is important because immunization doses can be limited or very expensive in practice, and without an optimal distribution these resources can go inadvertently to waste.

To investigate the applicability of CI to an immunization/quarantine scheme in a real large-scale social network we consider a social contact network built from the mobile phone

calls between people in Mexico. Data has been provided by GranData.

A mobile phone call social network reflects people's interactions in social lives, and is generally accepted as a proxy of a human contact network. For example, the mobile phone network from Mexico can help us to design effective immunization strategies, by identifying the most relevant social contacts among people. The disease spreads through direct contacts of infected people and proximity and mobility data from mobile phone networks can serve as a proxy of human movements and possible spreading patterns in human contact networks.

In order to build the network, we put a link between two people if there is a reciprocal exchange of phone calls between them in a observation window of three months (i.e. a call in both directions), and the number of such reciprocal calls is larger than or equal to three. This criterion gives us a network of $N = 14,346,653$ nodes, with an average degree $\langle k \rangle = 3.53$ and a maximum degree $k_{\max} = 419$. The result of the CI algorithm, compared to HD and HDA, is shown in Fig. 3d.

The phone call network is the prototype of big-data, where a scalable (i.e. almost linear) algorithm is mandatory. Indeed, the size of this network already rules out many heuristic methods with quadratic (or larger) running time (CC, EC, BC, and EGP) and also BP. From the perspective of performance, CI is better by a very good margin. Indeed, it fragments the network using about $500,000$ people less than the best heuristic strategy (HDA) implying a saving of the same number of vaccines in a hypothetical immunization campaign. Moreover, when CI gives a zero giant component, HDA gives still $G \sim 0.3$, i.e. a connected network of $\sim 4 \times 10^6$ people. This result, together with the result on Twitter, indicates that, although the theory has been developed for a locally-tree like network, in real networks with loops the CI-algorithm performs quite well as well.

[31] Colizza, C., Flammini, A., Serrano, M. A. & Vespignani, A. Detecting rich-club ordering in complex networks. *Nature Phys.* **2**, 110-115 (2006).

[32] Wasserman, S. & Faust, K. *Social Network Analysis* (Cambridge Univ. Press, Cambridge, 1994).

[33] Straffin, P. D. Linear algebra in geography: eigenvectors of networks. *Mathematics Magazine* **53**, 269-276 (1980).

[34] Bavelas, A. Communication patterns in tasks oriented groups. *J. Acoust. Soc. Am.* **22**, 271-282 (1950).

[35] Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry* **40**, 35-41 (1977).

[36] Mézard, M. & Montanari, A. *Information, Physics, and Computation* (Oxford University Press, USA, 2009).

[37] Centola, D., Eguiluz, V. & Macy, M. W. Cascade dynamics of complex propagation. *Physica A: Statistical Mechanics and its Applications* **374**, 449-456 (2007).

[38] Singh, P., Sreenivasan, S., Szymanski, B. K. & Korniss, G. Threshold-limited spreading in social networks with multiple initiators. *Sci. Rep.* **3**, 2330 (2013).

[39] Baxter, G. J., Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. F. Bootstrap percolation on complex networks. *Phys. Rev. E* **82**, 011103 (2010).

[40] Wormald, N. C. The asymptotic connectivity of labelled regular graphs. *J. Combinatorial Theory B* **31**, 156-167 (1981).

[41] Dorogovtsev, S. N., Mendes, J. F. F. & Samukhin, A. N. Metric structure of random networks. *Nucl. Phys. B* **653**, 307-338 (2003).

[42] Bhatia, N. P. & Szegö, G. P. *Stability theory of dynamical systems* (Springer-Verlag, Berlin Heidelberg, 2002).

[43] S̃ulc, P. & Zdeborová, L. Belief propagation for graph partitioning. *J. Phys. A: Math. Theor.* **43**, 285003 (2010).

[44] Parisi, G., Ritort, F. & Slanina, F. Several results on the finite-size corrections in the Sherrington-Kirkpatrick spin-glass model. *J. Phys. A: Math. Gen.* **26**, 3775-3790 (1993).

[45] Bray, A. J. & Moore, M. A. Chaotic nature of the spin-glass phase. *Phys. Rev. Lett.* **58**, 57-60 (1987).

[46] Ferguson, N. M., *et al.* Strategies for containing an emerging influenza pandemic in Southeast Asia. *Nature* **437**, 209-214 (2005).

[47] Auerbach, D. M., Darrow, W. W., Jaffe, H. W. & Curran, J. W. Cluster of cases of the acquired immune deficiency syndrome. Patients linked by sexual contact. *Am. J. Med.* **76**, 487-492. (1984).

[48] Guggiola, A., Semerjian, G. Minimal contagious sets in random regular graphs. *J. Stat. Phys.* **158**, 300-358. (2015).

[49] Zhou, H.-J. Spin glass approach to the feedback vertex set. *Eur. Phys. J. B* **86**, 455-463 (2013).

[50] Qin, S.-M. & Zhou, H.-J. Solving the undirected feedback vertex set problem by local search. *Eur. Phys. J. B* **87**, 273 (2014).