Morphological inversion of complex diffusion

V. A. T. Nguyen and D. C. Vural*

University of Notre Dame, Department of Physics, 225 Nieuwland Science Hall, Notre Dame, Indiana 46556, USA

(Received 1 February 2017; revised manuscript received 8 September 2017; published 26 September 2017)

Epidemics, neural cascades, power failures, and many other phenomena can be described by a diffusion process on a network. To identify the causal origins of a spread, it is often necessary to identify the triggering initial node. Here, we define a new morphological operator and use it to detect the origin of a diffusive front, given the final state of a complex network. Our method performs better than algorithms based on distance (closeness) and Jordan centrality. More importantly, our method is applicable regardless of the specifics of the forward model, and therefore can be applied to a wide range of systems such as identifying the patient zero in an epidemic, pinpointing the neuron that triggers a cascade, identifying the original malfunction that causes a catastrophic infrastructure failure, and inferring the ancestral species from which a heterogeneous population evolves.

DOI: 10.1103/PhysRevE.96.032314

A sugar piece placed in tea will erode and eventually dissolve. Given the initial shape of the piece, it is trivial to predict its final distribution. However, the opposite problem of determining the initial state, given a final one is extremely difficult. Problems of the latter kind are referred as ill-posed inverse problems [1–3].

Diffusion taking place on networks, in the forward direction, is well studied. One class of models originally used to describe epidemics is the susceptible-infected-recovered (SIR) model [4,5]. Variations include SI, SIS, SIRS, etc. Others include more realistic delay conditions, such as an incubation period for the infection [6]. Similar models are used to describe neural cascades [7], traffic jams [8], and infrastructure failures [9].

Accordingly, a successful method of inverting diffusion on complex networks can help identify patient zero in an epidemic outbreak, pinpoint neurons that trigger a cognitive cascades, remedy the parts of the road network that initiate congestion, and determine malfunctions that lead to cascading failures. In the weak selection limit, evolution can be thought of as diffusion on a genotype network [10,11], so diffusion inversion may be used to identify ancestral species.

Here we address the problem of identifying the origin of a diffusive process taking place on a complex network, given the its final state. We refer to the influenced nodes as the candidate set C. Any member of C may be the node from which the diffusion originated. We refer to this node as the seed, s, and to the forward model as M.

Presently, there are two approaches to identify s. The first uses probability marginals from Bayesian methods [12–21]. In some cases, it is possible to sample the state space using Monte Carlo simulations [13]. However, this is only feasible for small networks. Message-passing algorithms can approximate the marginals efficiently [12,14–16]; however, these algorithms are model specific: For every M, one must invent new approximations, heuristic assumptions, and analytic calculations.

In contrast, the second class of methods works independent of the forward model [14,17–19]. These presuppose that s should be approximately equidistant to all other nodes in C, and therefore, nodes with high "centrality" values should have a higher likelihood of being s. This assumption breaks down if the spread reaches "boundaries" or if the spread self-interacts (i.e., if the network contains many loops rather than being a tree or a dynamic like SIS).

Here we present a method that can determine the origin of a diffusive process taking place on a complex network, regardless of what the diffusion model is, without the drawbacks of centrality-based methods. We take as inputs the network structure, the candidate set C and the forward model M. In return, we output a list of nodes, ordered according to the likelihood of being the seed s. We emphasize that our method has no free parameters and is applicable to any M, including both deterministic and stochastic forward models.

To evaluate our success, we performed simulations in the forward direction using four types of forward models on three different graph topologies. We then inverted the final state and determined how often our guess is the true seed. We also measured the error distance, i.e., the distance of our guess from the true seed.

The forward models we explored are susceptible-infected (SI) epidemic model with uniform propagation probability between neighboring nodes; an information cascade (IC) model which propagates the diffusion like the SI model but with an additional cascade effect based on the fraction of infected neighbors [22]; a collective behavior (CB) model based on the notion that social behavior is determined by threshold for when the benefit of an action is greater than its cost [23]; a heterogeneous SI model where the propagation probability has a directional bias (DB) based on spatial positions of the nodes.

The network topologies on which we evaluate our model consist of a real power grid (GRID) network of the western states of the USA [24], a real protein (PROT) interaction network of *C. elegans* [25], and a synthetic scale-free (SCLF) network based on the power grid network.

The spread time was selected such that none of the networks tested was fully saturated by the spread. This allows the final state to retain some unique characteristics that can be used to identify the seed. We did not explore cases for low propagation probability and large spreads in order to maintain a consistent total simulated time, $T = 5$, for all models.

*dvural@nd.edu

I. GENERALIZED MORPHOLOGICAL OPERATORS

The principle behind our method can be best described by the language of mathematical morphology pioneered by Minkowski, Matheron, Serra, and others [26–28]. A morphological operator modifies every point in a set (e.g., an image) according to the spatial arrangement of neighboring points. A stencil, called the "structuring element," with a predefined shape is placed on individual points, and if the surroundings of the point match (or not match) the shape of the stencil, then the point is modified. One particular operator, erosion, is important for our purpose. Erosion deletes all points whose surroundings mismatch the structuring element. Since a mismatch would typically happen near the boundaries of a shape, the erosion filter ends up rounding up and thinning down all shapes. This is the qualitative behavior we need in order get rid of the peripheral nodes of C and reach its core.

To suit our specific purpose, we define a new morphological operator analogous to erosion, but with three important differences (Fig. 1). First, our structuring element is not fixed, but changes according to where it is placed on the network. Furthermore, our structuring element does not have sharp edges but is fuzzy. To be precise, we take the structing element, when placed on node i, to be $P(r,j|i)$, for node j in state r which we compute numerically.

Second, the comparison of the structuring element and the surrounding nodes of i is not binary but weighted. This is because mismatches of deterministic events (e.g., $P(r,j|i) \sim 0$ or 1) matter more than random events ($P(r,j|i) \sim 0.5$). To be precise, we weight every node mismatch with a factor inversely proportional to the binary entropy $H_b[P(r,j|i)]$.

Third, the final effect of processing a node with a structuring element is not simply deleting or keeping. Instead, this too is fuzzy. In the end, upon applying our morphological operator to the network once, we expect the least eroded node to be the seed.

Our algorithm generates an ordered list of candidate seeds based on how much they are eroded. The best-case scenario is when the true origin is located at the top of this list. Figure 1 schematically shows an evaluation of the match between an erosion stencil and the given network state along with an example of classical image erosion.

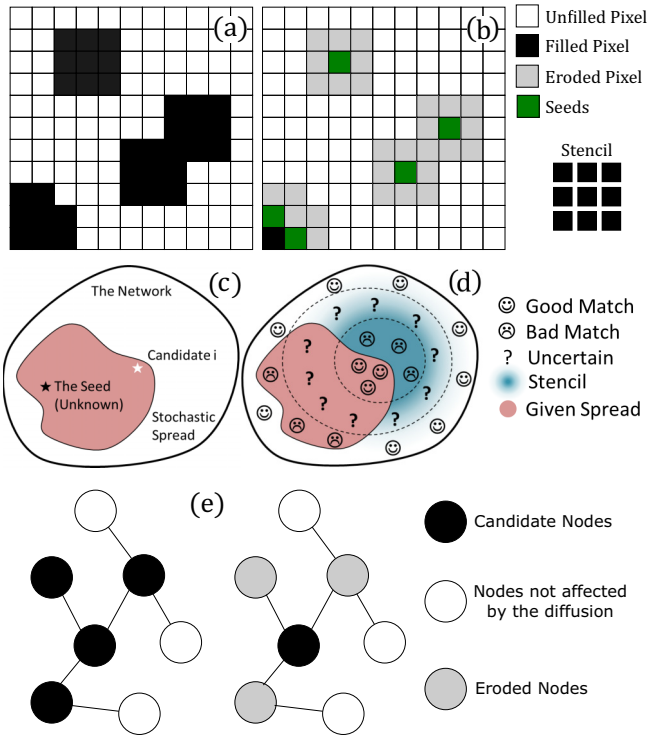

FIG. 1. Example of classical erosion (a,b), our generalization (c,d), erosion on a network (e). (a) The structuring element ("stencil") is placed on individual pixels whose neighborhoods are checked for a match or mismatch. (b) Pixels are eroded if there is a mismatch with the stencil. In the end, shapes lose their outer layers. (c) The network and candidate set (solid fill) is given while the seed (dark star) is unknown. (d) The erosion stencil (gradient fill) for candidate i (light star) is applied over the network. Smiley faces show the locations where the stencil matches the given network state, question marks show locations of high variability, and sad faces show regions where the stencil does not match the candidate set. Note that the planar representation for the stencils does not mean that our stencils are limited to planar graphs. All nodes are part of the stencil for any given node for a general diffusion dynamic. (e) An example of deterministic diffusion of a single seed for *one* discrete time step. The necessary stencil is one in which all neighbors are affected by the diffusion but no one else because there is not enough time to reach anything outside of direct neighbors. For this reason, the node in the top left was also eroded.

II. METHOD

The structuring elements are generated by directly sampling the states of the network model. For each forward model (defined explicitly in the next section), we applied the selected diffusion dynamics 500 times for every node in the network and calculated the structing element, $P(r,j|i)$, based on the normalized frequency of finding node j in state r due to the diffusion starting from node i.

We define a convergence condition for our stencils based on the average absolute error of the probabilities after moving to a higher sample size. For a sample size n, we require that the average error, δP, be less than 1% after moving to a sample size of $5n$:

$$\delta P = \max_{\lambda\in(0.1,0.5,0.95)} \sum_{i,j} |P_\lambda^{5n}(i|j) - P_\lambda^{n}(i|j)| \leqslant 0.01.$$

Naturally, the error will depend upon the diffusion parameters and the network topology. However, we found that it is reasonable to just sample the errors using the largest network for a subset of the diffusion parameters. For the most part, our algorithm converges very quickly where the final sample size for the stencils was 500 runs for each node in the network.

The idea behind our morphological filter is to determine which stencil has the best match with the given diffusion state. The likelihood of a node being the origin node is proportional to the similarity between its stencil and the diffusion state. However, there are many ways to measure the similarity between a probability vector and a binary vector. We sampled the performance of different scoring metrics such as log-likelihood (under independent three-body correlations), information surprisal, and Picard distance. The results shown

in this paper are based on the an information theoretical metric which we found to work best. For each node i inside the candidate set C of the given final network state, we apply an erosion score:

$$S_i = \sum_j \frac{1 - P(r_j, j|i)}{H_b(P(r_j, j|i))},$$

where the weight $H_b(x) = -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy for the two-state diffusion, r_j is the state of node j in the given final network state. In other words, S_i measures the mismatch between the stencil and diffusion state weighted by the (binary) entropy, H_b, of the probability distribution. This weight will diminish the value of nodes with high variability ($p \approx 0.5$) in comparison to nodes with low variability ($p \approx 0$ or $p \approx 1$). If the probability that a node is affected is very high or very low, then mismatches are weighted heavily and have a large negative influence on S_i. On the other hand, the state of highly variable nodes are circumstantial, and thanks to the small entropic weight they do not have much influence on S_i. Note that the score for candidate node i examines its stencil element, $P(r, j|i)$, at every other node j. In other words, the stencil for any node involves all other nodes in the network and not just the nodes in the set C.

Once we have S_i for all i, we sort these in ascending order and pick the nodes with best (i.e., lowest) scores. Numerically, the entropic weight can result in a division by zero and therefore instead of directly using $P(r, j|i)$ we used $P(r, j|i) + \epsilon$, where $\epsilon = 10^{-20}$ is small enough to not change the degree of variability and instead provides an upper bound for a highly unexpected mismatch. If the relevant forward dynamics is one where nodes can take more than two states, then the binary entropy function should be updated to be the entropy for the probability stencil of node i causing node j to be in state r:

$$H = -\sum_r P(r, j|i) \log_e P(r, j|i).$$

The error bars generated are based on the standard deviation of our results. For each network topology and diffusion model, we simulated 500 realizations of the diffusion in order to test the performance of our algorithm. We separated these into 5 sets of 100 simulations by random assignment. We then calculated our performance inside each of the sets separately and used their standard deviation for our errors. We then repeated the random assignment 100 times and averaged over all of the standard deviations. This removes the dependence on how the simulations were randomly assigned as well as provide a measure of error for a trial containing 100 simulations. Additionally, some realizations will not have a candidate set larger than one when the spread probability is very small. We simply remove these cases from our calculation and therefore the actual sample size for low λ ($\approx$250) is lower than for $\lambda > 0.80\%$ (500).

III. NETWORKS AND FORWARD DYNAMICS

To evaluate our inversion scheme, we used a protein-protein interaction network [25], a power grid network [24], and a synthetic scale-free network. Our diffusion dynamics are discrete in time and are fully described by the probability that an "infected" node spreads to a susceptible neighbor. The SI model is defined by the probability $p_{\mathrm{ij}} = \lambda A_{ij} I_j$ of an infection spreading from j to i, where A_{ij} is one when nodes i and j are connected by an edge and zero otherwise and I_j is one if node j is infected and zero otherwise, which simplifies to j infecting i with probability $\lambda = [0.05, 0.1, \ldots, 0.95]$ only if the two nodes are adjacent and j is infected.

The IC model cascades the information spread based on the state of a critical fraction of neighbors, $\nu = 0.5$, via

$$p_{ij} = \begin{cases} 1, & \sum_j A_{ij}(I_j - \nu) \geqslant 0, \\ \lambda A_{ij} I_j, & \sum_j A_{ij}(I_j - \nu) < 0. \end{cases} \quad (1)$$

The CB model spreads the adaptation of a social behavior when the number of neighbors who have adopted the behavior reaches an absolute threshold, $\mu = 2$, via

$$p_{ij} = \begin{cases} 1, & \sum_j A_{ij} I_j \geqslant \mu, \\ \lambda A_{ij} I_j, & \sum_j A_{ij} I_j < \mu. \end{cases} \quad (2)$$

The DB model uses heterogeneous diffusion probabilities $p_{ij} = B_{ij} I_j$, where $B_{ij} = A_{ij}(p_0 + \delta p \cos[\vec{d}_{ij} \cdot \vec{b}])$. The DB model is generated by first randomizing the three dimensional positions for all nodes placed uniformly random inside a unitary cubic volume and then calculating the unit displacement vector, $\vec{d}_{ij}$, between graphically ($A_{ij} = 1$) adjacent nodes. We then picked a unit bias vector, $\vec{b}$, pointing toward one of the corners of the volume, and generated the weighted adjacency matrix B_{ij} based on a neutral transmission probability $p_0 = [0.2, 0.4, 0.6, 0.8]$ and range $\delta p = 0.15$.

IV. DISTANCE AND JORDAN CENTRALITY

The origin of a diffusion process can *prima facie* be expected to be found near the "center" of the candidate set. Thus, we compare our results with two benchmark methods based on centrality measures (Fig. 2). Centrality-based methods calculate the distances between pairs of candidate nodes (i, j), D_{ij}, inside the *subgraph* generated by only their connections. This means that each diffusion will require a new calculation of the distances because the candidate nodes will generally never be the same set of nodes. The distance (closeness) centrality $\mathcal{D}_i$ of a node i refers to the total distance between node i and all other candidate nodes j:

$$\mathcal{D}_i = \sum_{j \in C} D_{ij}.$$

This method assumes that the node which has the least distance to all other candidate nodes is the most likely seed. The Jordan centrality $\mathcal{J}_i$ of a node i is concerned only with the largest distances between node i and all other candidate nodes.

$$\mathcal{J}_i = \max_{j \in C} D_{ij}.$$

Similar to the distance centrality, this method assumes that the most likely candidate node is the least distant to all other candidate nodes.

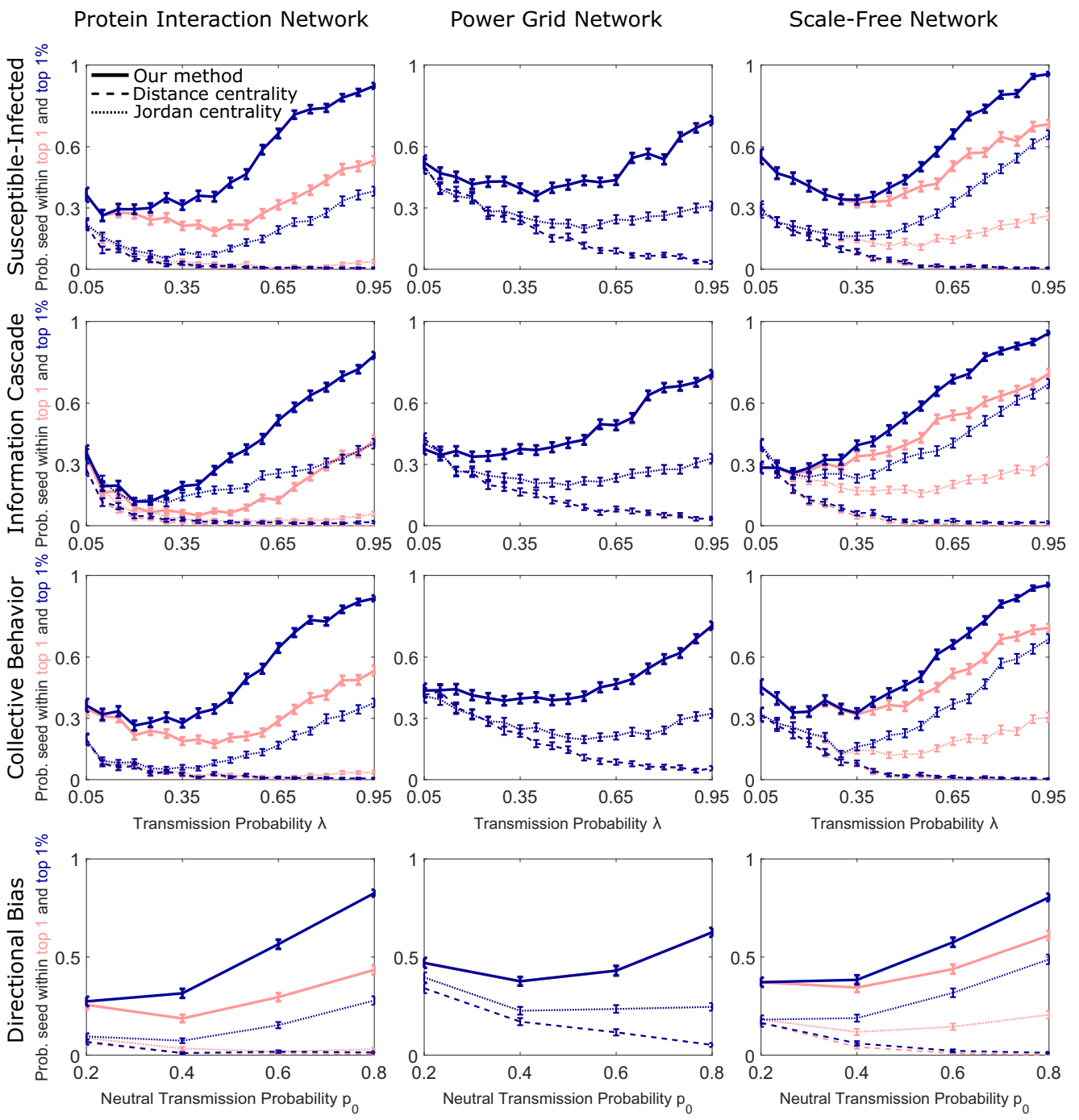

FIG. 2. Performance on different networks and models as a function of the transmission probability λ, for $T = 5$ (larger is better). The protein interaction network ($n = 3744, m = 7749$), power grid network ($n = 4941, m = 6594$), and scale-free network ($n = 4941, m = 6601$) are shown in each column. The four dynamic models are shown in each row. Note the difference between the uniform transmission probability, λ, for isotropic diffusion and the neutral transmission probability, p_0, for anisotropic diffusion. Our performance (solid lines) is almost always better than centrality-based methods (dashed and dotted lines of the same color) based on a total of 500 runs. Light red and dark blue curves denote whether the true seed is the top one or within top three choices returned, respectively. For the power grid network, both cases overlapped because the candidate set is small. The error bars represent $\pm$ one expected standard deviation for an average of 100 runs. In general, our performance closely matches the two centrality methods but becomes noticeably better as λ or p_0 increases.

V. NUMERICAL ALGORITHM

Given a graph $G(V,E)$, forward diffusion model, total time T, and realization R to be inverted, we enumerate the necessary steps to use our algorithm:

(1) Generate the candidate set C based on R. For SI dynamics, the set C contains every node which is in state I.

(2) For every node $i \in C$, apply the diffusion dynamic for total time T and repeat for a total of M independent trails. Record the probability that node $j \in V$ was in state r, $P(r,j|i)$.

(3) Calculate the erosion score for every node $i \in C$ via

$$S_i = \sum_j \frac{1 - P(r_j, j|i)}{H_b(P(r_j, j|i))}.$$

(4) Assign ranks to every candidate node based on their scores. The lowest score has the first rank and is the most likely candidate based on our erosion.

Our current scheme is based only on SI-type dynamics. For dynamics with additional states such as SIR or SIRS, the candidate set must be carefully considered. In the most simple case, the candidate set can be the entire graph. The erosion score must also use the entropy rather than the binary entropy.

VI. RESULTS

Many authors use distance error as a metric of success [13,14,19,21]; however, the usefulness of this metric is ambiguous. Although a two-hop range constitutes a small fraction of the network (0.2%,1%,5% for GRID, SCLF, and PROT, respectively), such small percentages still correspond to a significant absolute number of nodes (10, 50, and 210 nodes for GRID, SCLF, and PROT respectively). We provide the results for the distance error of our algorithm in the Appendix and here only focus on the probability of finding the true seed and the rank distribution of the true seed. All simulated runs were done on the Notre Dame Center for Research Computing's High Performance Computing clusters.

Figure 2 shows how often the true seed is our top guess and how often it is within our top three guesses. The bold solid lines show the performance of our algorithm, while the dashed and dotted lines show the performance of methods based on distance centrality and Jordan centrality. Our success rates are far above the dashed and dotted curves of the same color, with the only exception of the low-λ regime of the IC model.

Figure 3 show the ranking spectrum for the true seed using the three methods for $\lambda = 0.2$ ($p_0 = 0.2$) and $T = 5$. The protein and scale-free network has a heavy tail in comparison to the grid network because the spread can quickly reach many more nodes within $T = 5$ on the protein and scale-free network.

On average, the radius of C is $\langle r \rangle = T\lambda$ nodes. As $\lambda \to 0$, there are few nodes to pick from and thus centrality-based methods, including just randomly selecting a node from C, give similarly high success rates as our method. As λ is increased, however, the difference between our method and others increase significantly. The difference in success is maximal when the number of affected nodes become maximum, at $\lambda \to 1$. Across all forward models, we have the least success when the spread probability $\lambda \sim 0.5$, the regime with the highest number of possible states due to the high variability in the stochastic process. This variability causes the calculated average stencil to very rarely match a given diffusion state.

VII. DISCUSSION

In general, stochastic dynamics on networks will be defined in terms of local properties rather than global ones. To this extent, we explored two main variations to a local property, i.e., models in which fractional versus absolute number of affected neighbors determine the spread probability. We have also shown that our algorithm performs well for nonuniform and directionally biased diffusion. Hence, we have explored, nearly to full extent, the inversion of two-state diffusion processes on complex networks.

Our algorithm is a general method for determining the source of diffusion dynamics on complex networks. While the generation of morphological stencils requires knowing the dynamic law as well as the states of all nodes at some final time, our approach works for all models. In contrast, other inversion schemes are model specific [12,16,19,20]. Even though our inversion scheme works for any model, it is not model invariant like the centrality-based methods which uses only topological properties in the graph. The core part of our algorithm relies on an erosion of the diffusion surface by the diffusion stencils for each point in the spread. We need to know the specifics of the forward model in order to generate these diffusion stencils.

The performance of our algorithm may be improved by degeneralizing the scoring function to accommodate particularities of a forward model. We also note that there are many aspects of the problem we have not yet considered, such as cases of incomplete or noisy information, dynamics of multistate diffusion, and even multisource diffusion [12,13,15,16,20,21].

Such generalizations should be within reach: In the case of multisource, one could generate a scoring metric which is nonsymmetric against the diffusion state. In other words, the scoring could prioritize matching the diffusion state rather than the base state. In the case of multistate diffusion, one could introduce a transition matrix for the states where the elements of this matrix represents how easy or difficult it is to transition from one state to another. This matrix must be embedded into the scoring metric such that mismatches in state are weighted by the elements of this matrix. Additionally, some methods use a reduction scheme which considers more complex dynamics as two compartments. For example, Ref. [13] uses a SIR dynamic but then groups the status of I and R into a single compartment for their Jaccard similarity measure, which uses a binary status.

We now compare and contrast our method with another that is most similar in spirit to ours. Antulov-Fantulin *et al.* [13] uses a Jaccard similarity function to characterize the similarity between simulated spreads versus a given spread. This is then used to estimate the probability of a spread given a source via a Gaussian weighting and therefore we will refer to this method as the Jaccard-Gaussian algorithm. This method compares two network states, i.e., two binary vectors (nodes are either infected or susceptible) as obtained by Monte Carlo simulations versus a given final state. To compare two microstates of networks, Ref. [13] must generate, store, and compare all (or at least, most) possible realizations of

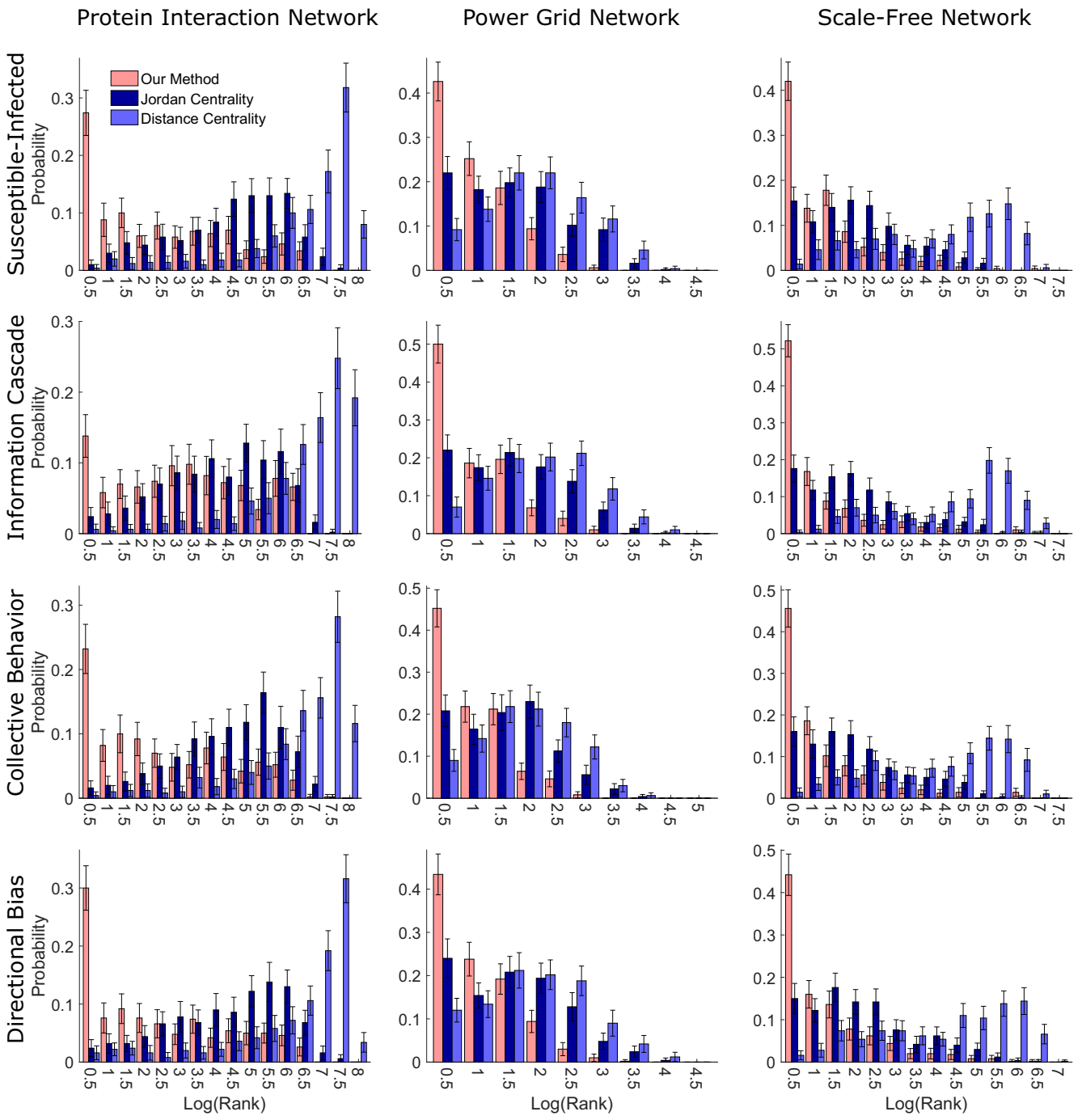

FIG. 3. Probability that the true seed is a high-ranked node for all models and dynamics using $\lambda = 0.6$ ($p_0 = 0.6$) $T = 5$ (lower rank is better). The protein interaction network ($n = 3744, m = 7749$), power grid network ($n = 4941, m = 6594$), and scale-free network ($n = 4941, m = 6601$) are shown in each column. The four dynamic models are shown in each row. The protein and scale-free networks have larger spreads than the grid network and therefore will have a larger distribution for the rank sizes. Our method (solid lines) generally outperforms the two centrality methods based on a total of 500 runs. Note that Jordan centrality provides much better performance than distance (closeness) centrality. The error bars represent $\pm$ one expected standard deviation for an average of 100 runs.

a spread from every single candidate node. In contrast, we work with a single probability distribution defined over the network. Since the space of all possible N-node states (which Ref. [13] samples) is astronomically larger than the space of *single* node states (which we sample), we can leverage this gain in computational cost to sample our space more accurately. Our approach has another advantage: A forward model uniquely determines a stencil, and once we have our stencils for a certain model, we can use it for multiple C sets, say, for different realizations of the same disease. Reference [13], on the other

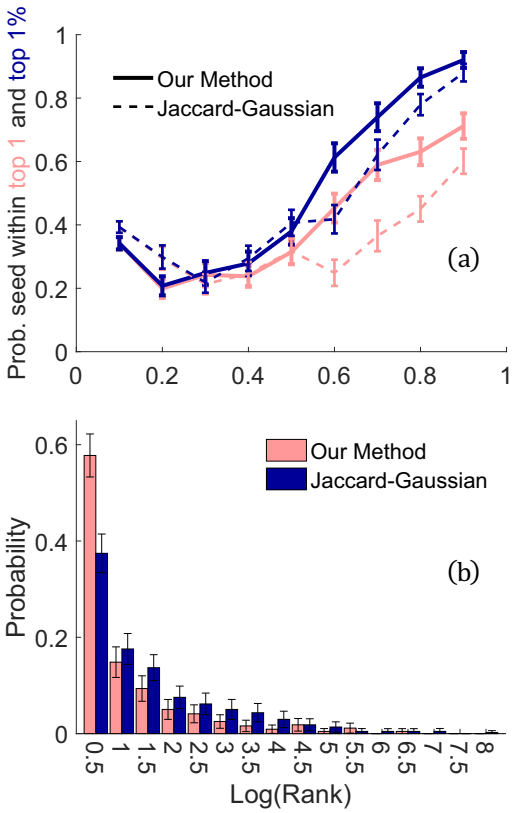

FIG. 4. Our method vs the Jaccard-Gaussian method developed by Antulov-Fantulin *et al.* in Ref. [13] on the scale-free graph ($n = 4941, m = 6601$). (a) Probability of finding the true seed within top 1 (light red) and top 1% (dark blue) candidate nodes. Overall, the Jaccard-Gaussian method has better performance when $\lambda < 0.5$, after which our method is better. In all cases, both methods were limited to using the same set of 134 simulated runs (stencil), where each data point was generated from independent test sets of 500 runs. However, only runs which satisfies the Jaccard-Gaussian convergence condition [Eq. (3)] were used in these plots. (b) How often a highly ranked node is the true seed ($\lambda = 0.7$, $T = 5$). Both methods used the same set (stencil) of 667 simulated runs for their algorithms. The spectrum is generated from testing the performance of each method for 500 runs. Additionally, the error bars were generated by randomly selecting 100 runs and calculating the deviation from average across all runs. Therefore, the error bars represent the expected standard deviation for a sample size of 100 runs.

hand, must realistically sample a combinatorially larger space for every realization of the spread. Another difference with Ref. [13] is in our scoring function; specifically, we use an entropic weight when comparing a single realization to a probability distribution. This allows us to decide which nodes to take more seriously than others.

We have implemented the Jaccard-Gaussian algorithm and its performance is plotted against our method in Fig. 4. The Jaccard-Gaussian algorithm relies upon a convergence condition in order to select the width parameter used in its Gaussian weighting. We follow the convergence condition defined in the supplementary material of Ref. [13]:

$$|P_n(\theta_{\text{map}}) - P_{2n}(\theta_{\text{map}})| \leqslant 0.05, \tag{3}$$

where P_n refers to the candidate probability distribution using a stencil of n simulated runs and θ_{map} refers to the most likely candidate node inside P_{2n}. Additionally, the two probability distributions were generated from independent simulations. We used the same set of potential Gaussian widths as Ref. [13] and selected the smallest weight for which the convergence condition is satisfied. We generated a stencil of 200 runs for different values of the transmission parameter λ and $T = 5$ and stored these runs for testing the performance of our algorithm against the Jaccard-Gaussian algorithm. In all cases, the available information is the same for both algorithms. However, the convergence condition for the Jaccard-Gaussian method limits the stencil set from 200 to 134. Therefore, we limit our algorithm's stencils to use the same 134 simulated runs as well.

To test how the performance is affected by the stencil size, we again generated a stencil set of 1000 runs for $\lambda = 0.7$ and $T = 5$. The convergence condition limits the final stencil used to 667 runs for both methods. The spectrum of the true seed's ranking in both methods are shown in Fig. 4(b). From Fig. 4(a), the probability of finding the true seed ($\lambda = 0.7$) for our method and Jaccard-Gaussian method is 0.5890 ± 0.0419 and 0.3653 ± 0.0422 respectively. When the stencil size is increased to 667 runs, these scores become 0.5776 ± 0.0430 and 0.3744 ± 0.0377 for our method and Jaccard-Gaussian method respectively. Although the Jaccard-Gaussian method should be improved with a larger stencil size, the amount of simulations required can be quite large.

We conclude our discussion with the limitations of our inversion scheme. As usual, there is a tradeoff between accuracy and generality. Our method should not be expected to perform better than methods that are custom tailored to specific models. By studying the specific dynamics, one can generate additional constraints and properties of the dynamics such as tree topology, exact analytical solutions, conservation laws, etc., that might aid in inversion [17,19]. Furthermore, the performance of our algorithm can be enhanced by extending what we have done with two-body correlations to n-body correlations or time-dependent correlations to calculate path integrals based on Bayesian inference (e.g., Ref. [12] for one specific model). However, generalizing such approaches for *any* forward model and *any* network topology is offset by the huge number of simulations required to resolve the correlations to within a useful error margin and will be very costly.

ACKNOWLEDGMENT

We thank Fatos Yarman Vural for her insights. This material is based upon work supported by the Defense Advanced Research Projects Agency, HR0011-16-C-0062.

APPENDIX

We calculate, for the convenience of comparison to other authors, the performance of our algorithm based on the distance between our top candidate and the true seed in Figs. 5, 6, and 7. A distance error of zero means that we correctly identified the true seed.

The average size of the diffusion is plotted for different topology, network, and transmission probabilities in Fig. 8.

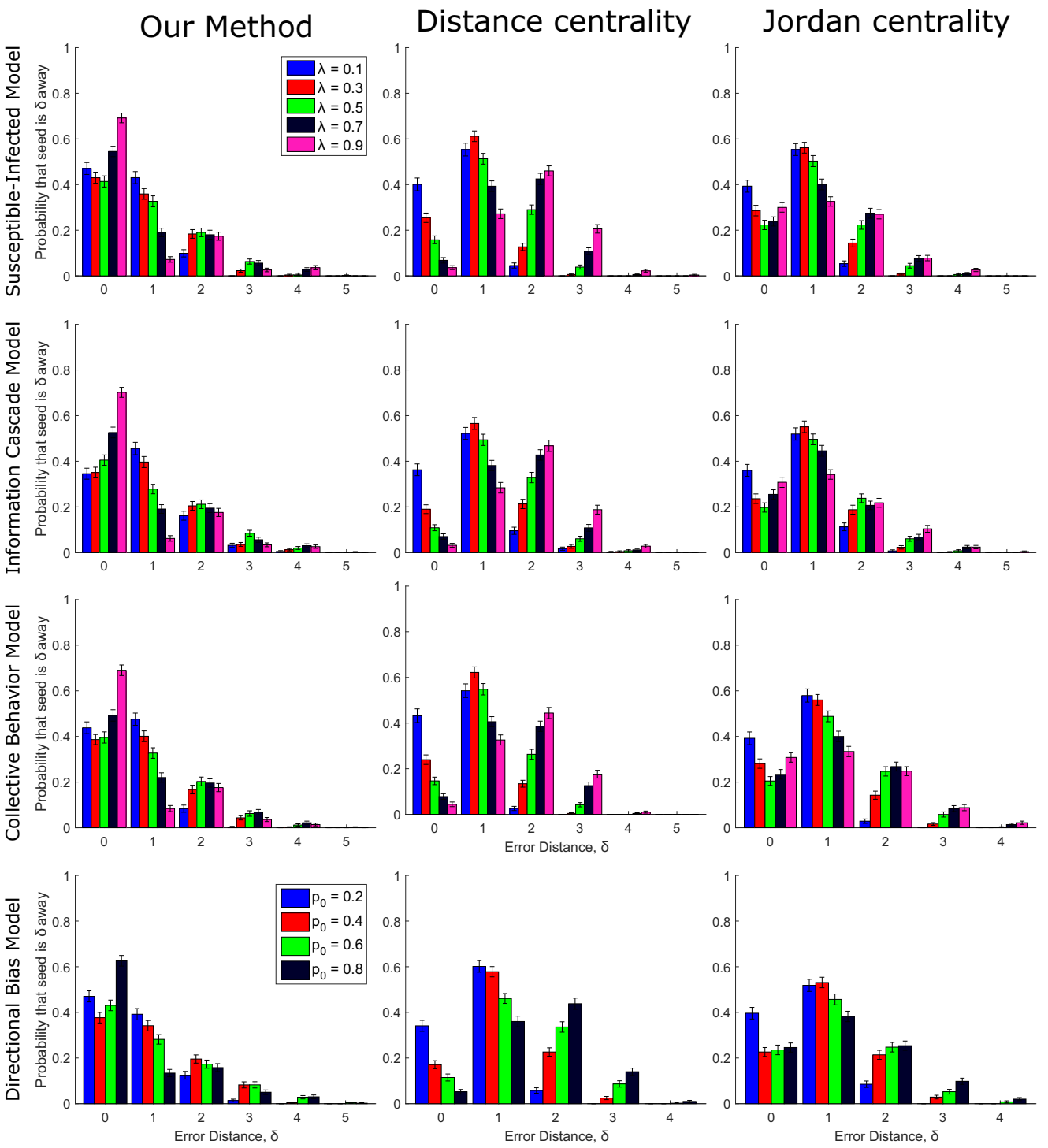

FIG. 5. Error distance δ for the power grid network ($n = 4941$, $m = 6594$). δ is the distance between the top candidate and the true origin.

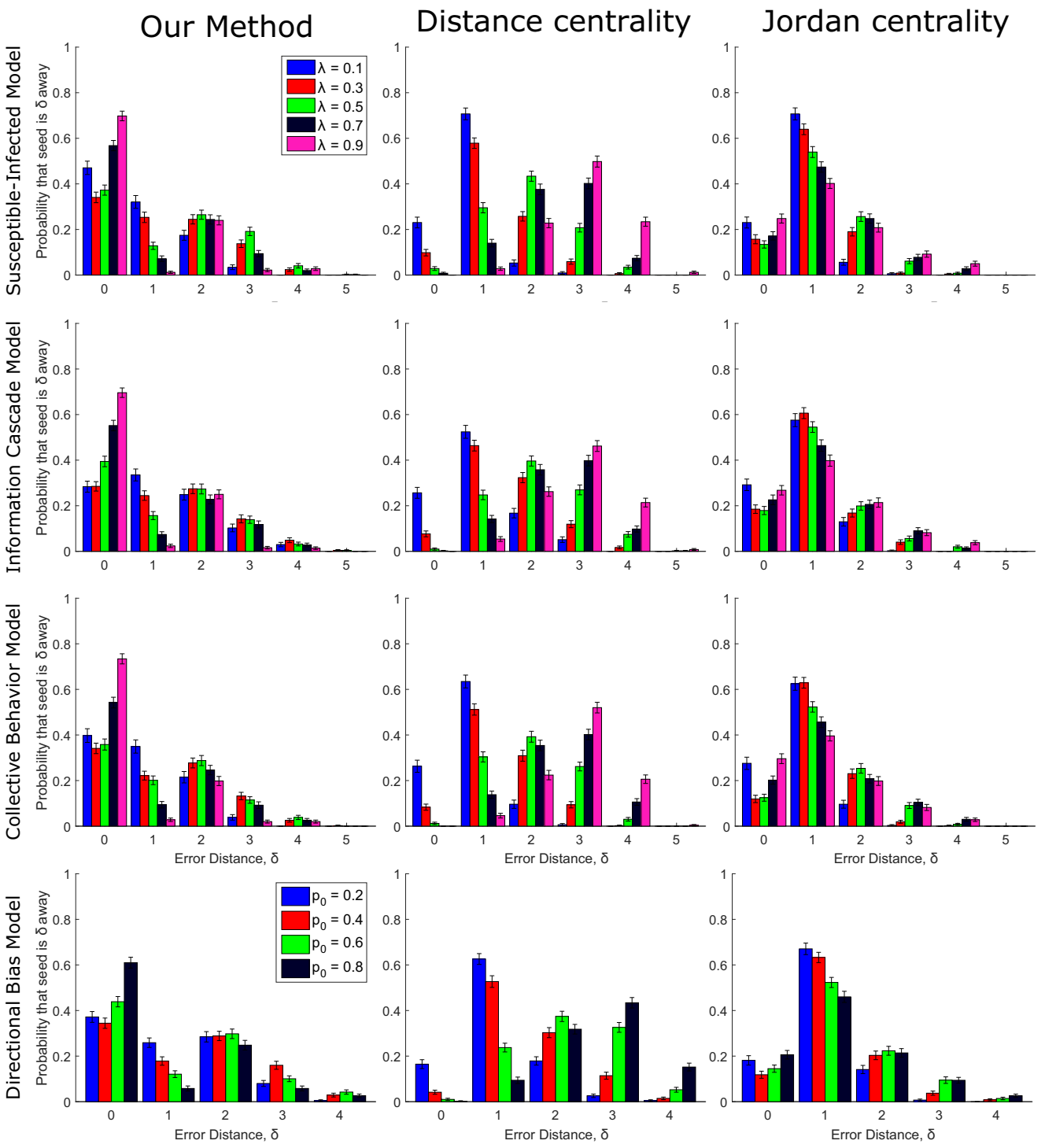

FIG. 6. Error distance δ for the scale-free network ($n = 4941$, $m = 6601$). δ is the distance between the top candidate and the true origin.

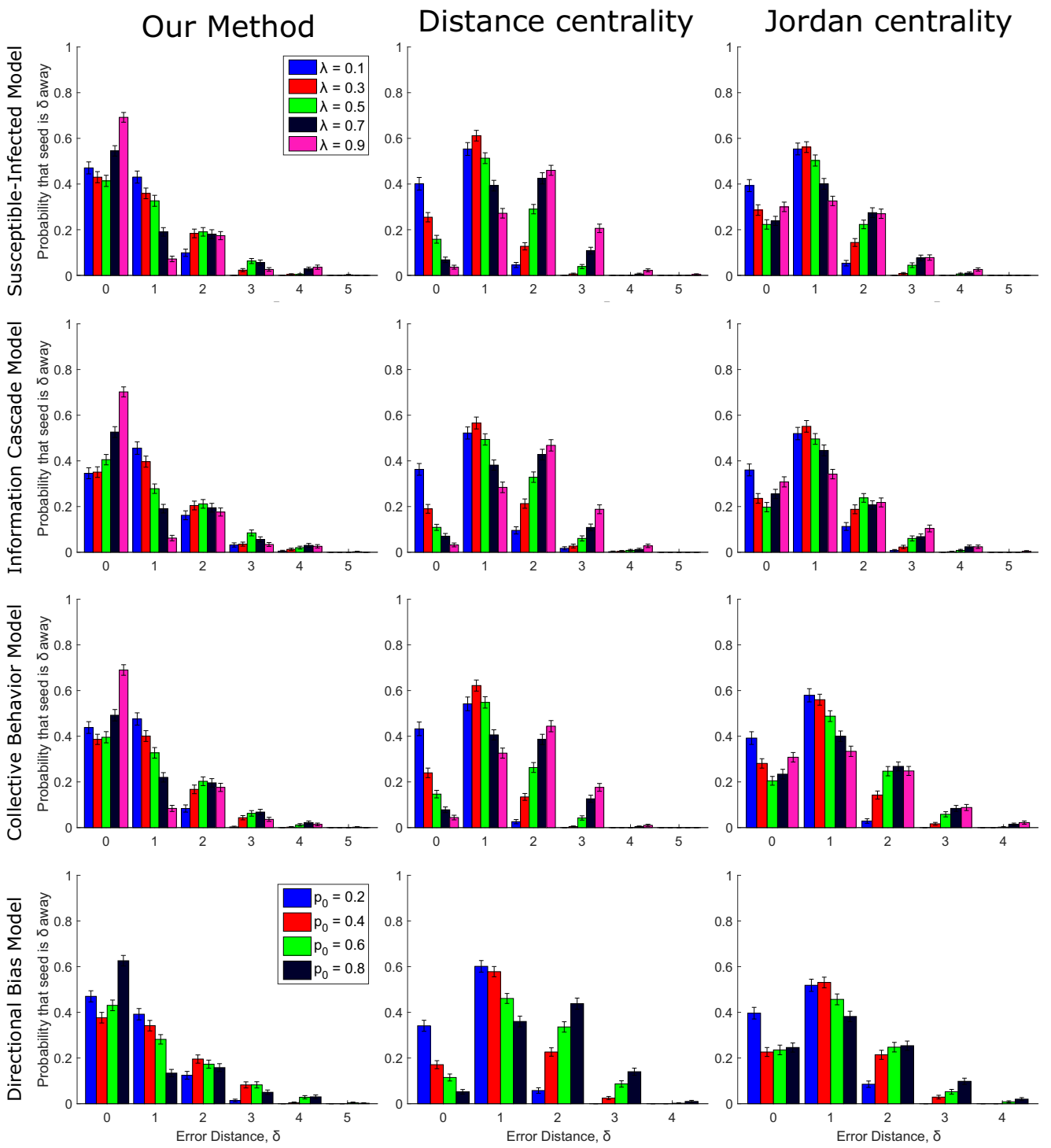

FIG. 7. Error distance δ for the protein interaction network ($n = 3744$, $m = 7749$). δ is the distance between the top candidate and the true origin.

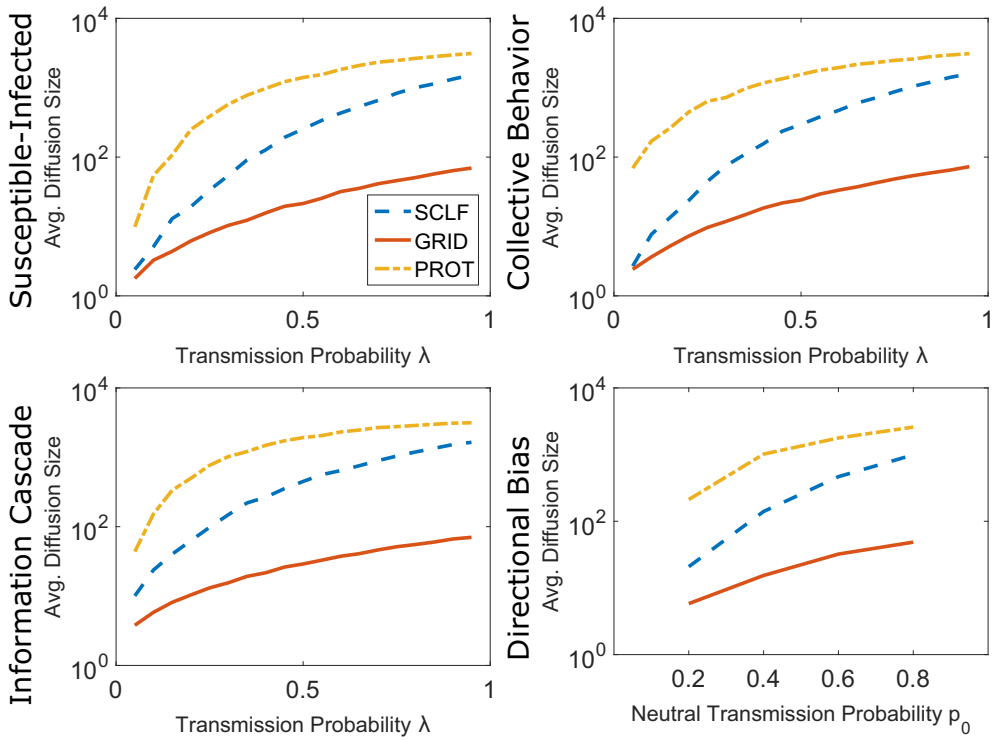

FIG. 8. Average diffusion sizes.

[1] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation* (SIAM, Philadelphia, 2005).

[2] C. Soussen, J. Idier, D. Brie, and J. Duan, From Bernoulli-Gaussian deconvolution to sparse signal restoration, IEEE Trans. Signal Process. **59**, 4572 (2011).

[3] P. Ruiz, X. Zhou, J. Mateos, R. Molina, and A. K. Katsaggeslos, Variational Bayesian blind image deconvolution: A review, Digit Signal Process **47**, 116 (2015).

[4] M. J. Keeling and K. T. Eames, Networks and epidemic models, J. R. Soc., Interface **2**, 295 (2005).

[5] W. O. Kermack and A. G. McKendrick, A contribution to the mathematical theory of epidemics, Proc. Math. Phys. Eng. Sci. **115**, 700 (1927).

[6] J. Zhen, Z. Ma, and M. Han, Global stability of an SIRS epidemic model with delays, Acta Math. Sci. **26**, 291 (2006).

[7] J. Wang and J. G. Lu, Global exponential stability of fuzzy cellular neural networks with delays and reactiondiffusion terms, Chaos Solitons Fractals **38**, 878 (2008).

[8] D. A. Kurtze and D. C. Hong, Traffic jams, granular flow, and soliton selection, Phys. Rev. E **52**, 218 (1995).

[9] S. Arianos, E. Bompard, A. Carbone, and F. Xue, Power grid vulnerability: A complex network approach, Chaos **19**, 013119 (2009).

[10] M. A. Huynen, Exploring phenotype space through neutral evolution, J. Mol. Evol. **43**, 165 (1996).

[11] C. O. Wilke, Adaptive evolution on neutral networks, Bull. Math. Biol. **63**, 715 (2001).

[12] F. Altarelli, A. Braunstein, L. DallAsta, A. Lage-Castellanos, and R. Zecchina, Bayesian Inference of Epidemics on Networks via Belief Propagation, Phys. Rev. Lett. **112**, 118701 (2014).

[13] N. Antulov-Fantulin, A. Lancic, T. Smuc, H. Stefancic, and M. Sikic, Identification of Patient Zero in Static and Temporal Networks: Robustness and Limitations, Phys. Rev. Lett. **114**, 248701 (2015).

[14] D. Shah and T. Zaman, Detecting sources of computer viruses in networks: Theory and experiment, ACM Sigmetrics Perform. Eval. Rev. **38**, 203 (2010).

[15] F. Altarelli, A. Braunstein, L. DallAsta, A. Ingrosso, and R. Zecchina, The patient-zero problem with noisy observations, J. Stat. Mech. (2014) P10016.

[16] A. Y. Lokhov, M. Mezard, H. Ohta, and L. Zdeborova, Inferring the origin of an epidemic with a dynamic message-passing algorithm, Phys. Rev. E **90**, 012801 (2014).

[17] W. Dong, W. Zhang, and C. W. Tan, Rooting out the rumor culprit from suspects, in *Proceedings of the IEEE International Symposium Information Theory* (IEEE, Piscataway, NJ, 2013), Vol. 2671.

[18] D. Shah and T. Zaman, Rumor centrality: A universal source detector, ACM Sigmetrics Perform. Eval. Rev. **40**, 199 (2012).

[19] D. Shah and T. Zaman, Rumors in a network: Who's the culprit? IEEE Trans. Inf. Theory **57**, 5163 (2011).

[20] W. Hu, W. P. Tay, A. Harilal, and G. Xiao, Network infection source identification under the SIRI model, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, NJ, 2015), Vol. 1712.

[21] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, Rumor source identification in social networks with time-varying topology, IEEE Trans. Dependable Secure Comput. **PP**, 1-1 (2016).

[22] D. J. Watts, A simple model of global cascades on random networks, Proc. Natl. Acad. Sci. USA **99**, 5766 (2002).

[23] M. Granovetter, Threshold models of collective behavior, Am. J. Sociol. **83**, 1420 (1978).

[24] D. J. Watts and S. Strogatz, Colective dynamics of "small-world" networks, Nature (London) **393**, 440 (1998).

[25] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, BioGRID: A general repository for interaction datasets, Nucl. Acids Res. **34**, D535 (2005).

[26] J. Serra, *Image Analysis and Mathematical Morphology* (Academic Press, San Diego, 1983).

[27] R. M. Haralick, S. R. Sternberg, and X. Zhuang, Image analysis using mathematical morphology, IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-9**, 532 (1987).

[28] P. K. Ghosh, A mathematical model for shape description using Minkowski operators, Comput. Vis. Graph. Image Process. **44**, 239 (1988).