

## Power-law scaling for the adiabatic algorithm for search-engine ranking

Adam Frees,<sup>1</sup> John King Gamble,<sup>2</sup> Kenneth Rudinger,<sup>2</sup> Eric Bach,<sup>3</sup> Mark Friesen,<sup>2,\*</sup> Robert Joynt,<sup>2</sup> and S. N. Coppersmith<sup>2,†</sup>

<sup>1</sup>*Department of Physics, Brown University, Providence, Rhode Island 02912, USA*

<sup>2</sup>*Department of Physics, University of Wisconsin-Madison, Madison, Wisconsin 53706, USA*

<sup>3</sup>*Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin 53706, USA*

(Received 11 December 2012; published 5 September 2013)

An important method for search engine result ranking works by finding the principal eigenvector of the “Google matrix.” Recently, a quantum algorithm for generating this eigenvector as a quantum state was presented, with evidence of an exponential speedup of this process for some scale-free networks. Here we show that the run time depends on features of the graphs other than the degree distribution, and can be altered sufficiently to rule out a general exponential speedup. According to our simulations, for a sample of graphs with degree distributions that are scale-free, with parameters thought to closely resemble the Web, the proposed algorithm for eigenvector preparation does not appear to run exponentially faster than the classical case.

DOI: [10.1103/PhysRevA.88.032307](https://doi.org/10.1103/PhysRevA.88.032307)

PACS number(s): 03.67.Ac, 03.67.Lx, 89.20.Hh

### I. INTRODUCTION

Quantum algorithms, which run on quantum computers, are known to be able to outperform classical algorithms for certain computational problems [1,2]. Thus, finding a new algorithm that exhibits a quantum speedup, in particular an exponential speedup, is of great interest [3]. An extremely important problem in computer science is calculating ranking for search engine results. PageRank, first proposed by Brin and Page [4], underlies the success of the Google search engine [5]. In this algorithm, websites are represented as nodes on a network graph, connected by directed edges that represent links. The matrix of network connections is constructed, and the PageRank vector is its principal eigenvector. Currently, computing the PageRank vector requires a time  $O(n)$ , where  $n$  is the number of websites in the network considered (e.g., the World Wide Web) [6]. Obtaining a quantum algorithm for PageRank that runs exponentially faster than the classical algorithm would be of great interest.

Recently, Garnerone, Zanardi, and Lidar (GZL) proposed an adiabatic quantum algorithm [7] to prepare the PageRank vector for a given network [6]. Remarkably, GZL present evidence that this algorithm can prepare the PageRank vector in time  $O[\text{polylog}(n)]$ , exponentially faster than classical algorithms for certain networks. It is important to note that the algorithm only offers an exponential speedup in the preparation of the PageRank vector; the information must then be extracted from the quantum state, which ultimately yields a polynomial quantum speedup. This polylogarithmic run time is due to the apparent logarithmic scaling of the gap between the two smallest eigenvalues of the Hamiltonian used in the algorithm (the energy gap). This scaling emerged on graphs constructed using adapted versions of two established methods of network construction: the preferential attachment model [8] and the copying model [9]. Both of these models yield graphs that are similar to the connectivity of the World Wide Web in that they are sparse (the total number of edges scales at most proportionally to the number of nodes) and scale-free

(the probability of finding a node with a specified in or out degree scales as a power law in those degrees). These features lead to networks that exhibit large-scale structure similar to that of the internet, such as being small world [10] and loosely hierarchical [11]. GZL studied sets of networks that exhibited both logarithmic scaling and polynomial scaling of the gap in the system size. However, they did not demonstrate that the networks with the favorable logarithmic gap scaling are scale-free over the region studied numerically.

Here we study the scaling of the GZL algorithm for graphs with degree distributions consistent with the internet. A realistic network model of the World Wide Web must be scale-free in both the in and the out degree [12,13]. We consider a broad variety of scale-free networks constructed by different methods. Choosing three well-known models for constructing random, scale-free networks, we control for both the mean degree and the exponent of the power law governing the degree distribution. We find that graphs with the same degree distribution can have different energy gap and run-time behaviors. Finally, we focus on degree distributions described by power laws consistent with those measured for the Web, both for the in degree and the out degree. We find that the relevant energy gap scales as a power of the system size, rather than logarithmically. These results demonstrate that for scale-free graphs with parameters thought to resemble the Web, the GZL adiabatic algorithm does not appear to yield an exponential quantum speedup for preparing the PageRank vector compared to current classical algorithms.

### II. NETWORK GROWTH MODELS

We generate samples of graphs with prescribed degree distributions using three different network growth models. GZL [6] use modified versions of two network construction algorithms: the preferential attachment model [8] and the copying model [9]. In addition to these two models, here we include also the more complex  $\alpha$ -preferential attachment model described by Bollobás *et al.* [12,14]. All three models grow random networks using probabilistic rules at discrete construction steps, which are detailed in Fig. 1.

All three of these models produce sparse, scale-free directed networks, in which the probability of the in degree (the number

\*friesen@physics.wisc.edu

†snc@physics.wisc.edu

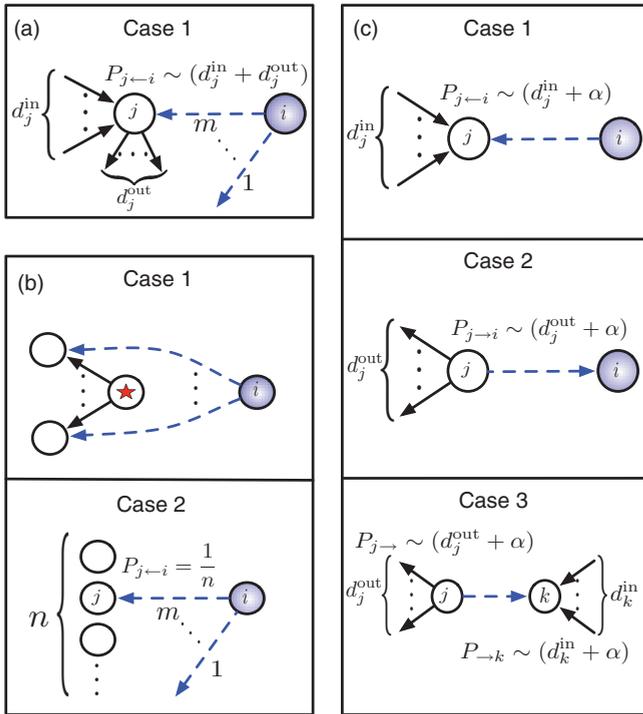


FIG. 1. (Color online) Illustrations of the three network generation models used. (a) GZL [6] preferential attachment, (b) GZL copying, and (c)  $\alpha$ -preferential attachment [12,14]. In all three models, a network is constructed by adding vertices and edges sequentially. (a) At each time step a new vertex  $i$  is added with  $m$  outgoing edges. The probability that one of these edges connects to a node  $j$  is proportional to the total degree of  $j$ . (b) At each time step there are two possible actions. With probability  $(1-p)$ , the new vertex points to all of the same vertices as the “star vertex,” which is a pre-existing vertex chosen uniformly at random at each time step. With probability  $p$ ,  $m$  outgoing edges are added to the new vertex, each pointing to vertices chosen uniformly at random. (c) There are three possible actions at each time step. With probability  $p_1$ , a new vertex is added with a single outgoing edge, pointing to a node  $j$  with probability proportional to the in degree of  $j$  plus a parameter  $\alpha$ . With probability  $p_2$ , a new vertex is added with a single incoming edge, pointing from a node  $j$  with probability proportional to the out degree of  $j$  plus  $\alpha$ . With probability  $(1-p_1-p_2)$ , no vertex, only an edge, is added. Its ending and starting points are determined as in cases 1 and 2, respectively. In all panels, the newly added edges are indicated by dashed lines.

of incoming edges) and out degree (the number of outgoing edges) of node  $i$  being equal to  $k$  are each proportional to a power law:

$$P[d_{\text{in}}(i) = k] \sim k^{-\gamma_{\text{in}}}, \quad (1)$$

$$P[d_{\text{out}}(i) = k] \sim k^{-\gamma_{\text{out}}}, \quad (2)$$

where  $d_{\text{in}}(i)$  and  $d_{\text{out}}(i)$  are the in and out degrees of node  $i$ , respectively, and the exponents  $\gamma_{\text{in}}$  and  $\gamma_{\text{out}}$  are typically between 2 and 4 [8]. The GZL versions [6] of the preferential attachment and copying models [8,15] produce networks that are scale-free in the limit of large graph size. However, due to the addition procedure described below, the networks are not necessarily scale-free for the sizes of graphs studied

numerically here and in Ref. [6]. To achieve networks that are scale-free in the out degree, GZL suggest to construct two networks,  $X$  and  $Y$ , independently.  $X$  and  $Y$  are each generated as in Fig. 1, except that for  $Y$  the direction of the edges added is reversed. The networks can then be added together, and the weights and loops discarded [6,16]. The resulting composite network is scale-free in both in degree and out degree, provided  $X$  and  $Y$  have the same number of edges per node. (See Appendix A for details.) In contrast to Ref. [6], the graphs studied here are all constrained in this way. However, the graphs exhibiting logarithmic scaling in [6] are not so constrained [16], and so they do not exhibit truly scale-free degree distributions over the numerically studied region. On the other hand, the  $\alpha$ -preferential attachment model (considered here but not in [6]) constructs a network which is scale-free in both in and out degrees without requiring an additional combination step. As with the GZL preferential attachment model, all weights and loops are removed from the final  $\alpha$ -preferential attachment network.

The exponents  $\gamma_{\text{in}}$  [Eq. (1)] and  $\gamma_{\text{out}}$  [Eq. (2)] of the degree distribution are model dependent. In the GZL preferential attachment model the number of edges added at each construction step controls the sparsity, and it is always the case that  $\gamma_{\text{in}} = \gamma_{\text{out}} = 3$  [8]. Both the GZL copying model and  $\alpha$ -preferential attachment allow for independently tunable exponents and mean degree. (See Appendix A for details.) This flexibility enables us to create three ensembles of model networks that have nearly identical degree distributions for  $\gamma_{\text{in}} = \gamma_{\text{out}} = 3$ . Furthermore, the last two models can be set with the exponents estimated for the World Wide Web [9,12], namely  $\gamma_{\text{in}} = 2.1$  and  $\gamma_{\text{out}} = 2.72$  [13].

### III. ALGORITHM DESCRIPTION

The Google matrix is constructed by taking as input an unweighted, simple network with  $n$  nodes [4], and representing it as an adjacency matrix  $A$ , where  $A(i,j) = 1$  if a directed edge points from node  $i$  to node  $j$ , and 0 otherwise. From this, one defines the matrix  $P$ :

$$P(i,j) = \begin{cases} 1/d_{\text{out}}(i) & \text{if } A(i,j) = 1, \\ 1/n & \text{if } \forall j, A(i,j) = 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3a)$$

$$P(i,j) = \begin{cases} 1/d_{\text{out}}(i) & \text{if } A(i,j) = 1, \\ 1/n & \text{if } \forall j, A(i,j) = 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3b)$$

$$P(i,j) = \begin{cases} 1/d_{\text{out}}(i) & \text{if } A(i,j) = 1, \\ 1/n & \text{if } \forall j, A(i,j) = 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3c)$$

The matrix  $P$  is stochastic because  $\sum_j P(i,j) = 1$  for all  $i$ .  $P$  can be thought of as a random walk (i.e., a web surfer), where the walker follows the network with equal likelihood of traversing all allowed links. If the walker ever reaches a dangling node (a node with  $d_{\text{out}} = 0$ ), Eq. (3b) implies that it can randomly hop to any vertex with equal probability. To prevent the walker from becoming trapped in an isolated portion of the network (a sink), the probability  $(1-\alpha_g)$  of moving to a node uniformly at random (including the possibility of staying still) is included, where  $0 < \alpha_g < 1$ ; Google uses  $\alpha_g = 0.85$ , which we also use here [6]. The Google matrix  $G$  is defined as the transpose of this resulting transition matrix:

$$G = \alpha_g P^T + (1 - \alpha_g) J, \quad (4)$$

where  $J$  is the matrix of all ones. The PageRank vector  $\vec{p}$  is the unique eigenvector associated with the largest eigenvalue

of  $G$ , which is 1. The run time of the best classical algorithm, which calculates the PageRank vector via power iteration, is  $O(n)$  [4,6].

To formulate an adiabatic quantum algorithm, GZL construct the Hamiltonian  $h(G)$ :

$$h(G) = (\mathbb{I} - G)^\dagger (\mathbb{I} - G), \quad (5)$$

which is Hermitian, even though  $G$  is not. The ground state of this Hamiltonian is the normalized PageRank vector. The adiabatic algorithm is completely defined by the interpolation Hamiltonian  $H(s) = sh(G) + (1-s)h(G_c)$ , where  $s \in [0, 1]$ , and  $G_c$  is the Google matrix for the complete network (including loops), whose ground state is a uniform superposition. The adiabatic theorem guarantees that if we initialize our system in the ground state of  $h(G_c)$  and change  $s$  from 0 to 1 sufficiently slowly, the system remains in the ground state [7]. Since the PageRank vector is the ground state of  $H(1) = h(G)$ , the PageRank vector is obtained when  $s = 1$ . The required slowness is also determined by the adiabatic theorem: As long as  $s(t)$  is a smooth function of the time  $t$  with  $0 \leq t \leq T$ , the run time  $T \sim \delta^{-b}$ , where  $b$  is  $O(1)$  and  $\delta$  is the energy gap between the ground and first excited state of  $H(s)$ , minimized over  $s$  [7]. Thus, an exponential speedup over the classical case is possible if  $\delta^{-1}$  is  $O[\log(n)]$ , since then  $T$  is  $O[\text{polylog}(n)]$ .

#### IV. NUMERICAL RESULTS

To study the scaling of the minimum energy gap  $\delta$  with the network size  $n$ , we compute  $\delta$  for the GZL Hamiltonian  $H(s)$ , averaging the results over many network realizations (typically 1000). Specifically, we calculate the minimum value of  $\delta$  over  $s \in [0, 1]$  using the Nelder-Mead method [17], where each objective function call calculates directly the eigenvalue spectrum of  $H(s)$ . We find that for most, but not all, network choices the minimum gap occurs when  $s = 1$ . Since  $H(s)$  is a dense matrix, this process is computationally intensive. We use the University of Wisconsin-Madison Center for High Throughput Computing and Open Science Grid to perform the simulations.

To assess whether the inverse energy gap  $\delta^{-1}$  scales logarithmically or as a power law in  $n$ , we plot in Fig. 2  $\delta^{-1}$  versus the network size on both log-linear and log-log scales, with data for the GZL preferential attachment, GZL copying, and  $\alpha$ -preferential attachment models. The model parameters are tuned (see Appendix A) so that all three have  $\gamma_{\text{in}} = \gamma_{\text{out}} = 3$  and have an average of two in and two out edges per node. Despite having nearly identical degree distributions [shown in Figs. 2(e) and 2(f)], the scaling of  $\delta^{-1}$  depends significantly on the method used to construct the graphs when viewed in Fig. 2(a). In Fig. 2(c) we show the distribution corresponding to the final data points in Fig. 2(a), where we see that the distributions are well separated and hence the construction models give different values of  $\delta^{-1}$ . By contrast, the degree distributions are difficult to distinguish, as shown in Fig. 2(d). Finally, we conclude that for all three models, the data are more consistent with  $\delta^{-1}$  scaling as a power law or a high-order polylogarithm, rather than a logarithm, as consistent with the data presented by GZL in the supplemental information of Ref. [6].

We next perform a similar analysis for degree distributions more closely related to the network of primary interest, the

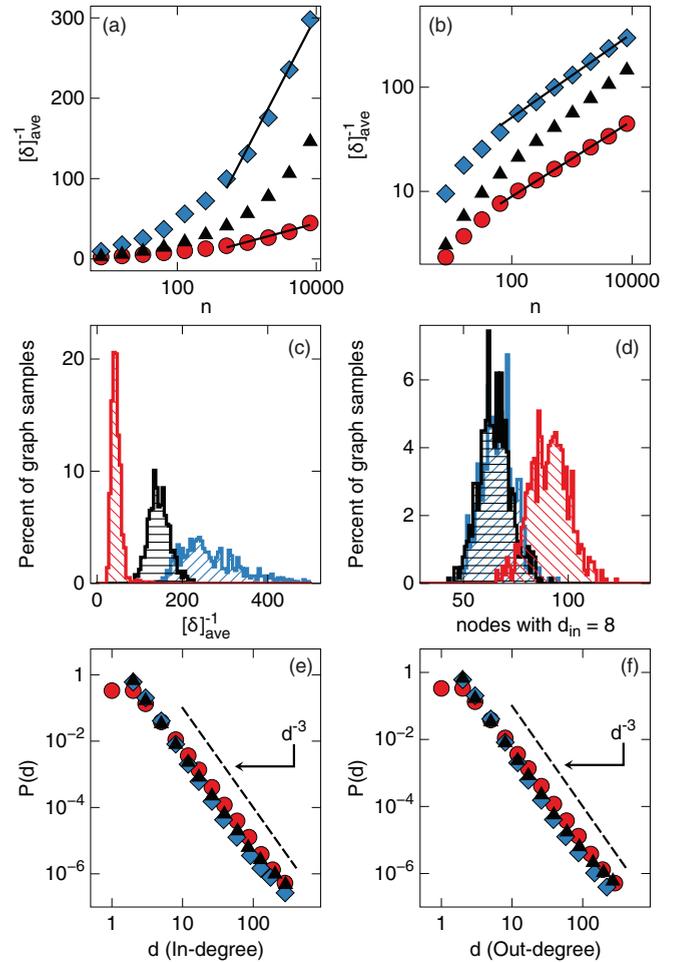


FIG. 2. (Color online) Comparison of the scaling of the inverse energy gap  $\delta^{-1}$  for the GZL [6] preferential attachment model (triangles, horizontal hatching), GZL copying model (diamonds, upward-sloping hatching), and  $\alpha$ -preferential attachment model [12] (circles, downward-sloping hatching), shown on (a) semilog and (b) log-log scales, demonstrating that  $\delta^{-1}$  is not proportional to  $\log(n)$  for these models. Results are averaged over 1000 random instances for  $n < 8192$ , and over 500 random instances at  $n = 8192$ . The fitting lines showed in (a) are  $72.2 \ln(n) - 363$  for the copying model and  $10.1 \ln(n) - 48.8$  for the  $\alpha$ -preferential attachment model. In (b), the fits shown are  $8.0n^{0.4}$  for the copying model and  $1.7n^{0.4}$  for the  $\alpha$ -preferential attachment model. If we fit the data instead to a power of a logarithm (not shown), we obtain  $0.56 \ln^{2.9}(n)$  for the copying model and  $0.18 \ln^{2.5}(n)$  for the  $\alpha$ -preferential attachment model. (c) Histogram of the inverse energy gaps for the data shown in (a) and (b) at  $n = 8192$ . (d) Histogram showing the distribution of number of vertices with in degree  $d_{\text{in}} = 8$  for  $n = 8192$ . (e) and (f) Degree distributions of the three models, demonstrating scale-free behavior and indicating that  $\gamma_{\text{in}} = \gamma_{\text{out}} = 3$ . Adaptive binning was used, as described in Appendix C. In all cases, both the mean in and out degree of each graph are two edges per node. These results demonstrate that  $\delta^{-1}$  differs significantly for the different graph construction methods, while the degree distributions are very similar.

World Wide Web, for which a realistic set of degree parameters is given by  $\gamma_{\text{in}} = 2.1$  and  $\gamma_{\text{out}} = 2.72$  [13]. As mentioned above, the preferential attachment model cannot be tuned to

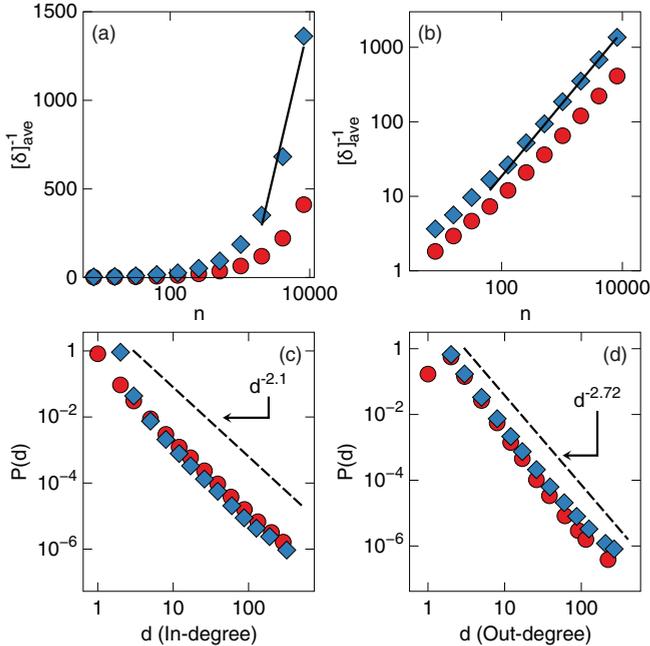


FIG. 3. (Color online) Inverse energy gap scaling for GZL [6] copying model (diamonds), and  $\alpha$ -preferential attachment model [12] (circles) of WWW-like networks, shown on (a) semilog and (b) log-log scales. Results are averaged over 1000 random instances for  $n < 8192$ , and over 500 random instances at  $n = 8192$ . In (a) the line fit shown is  $730 \ln(n) - 5300$ , while in (b) the line fit is  $0.2n^{0.97}$ . If we fit the data to a power of a logarithm (not shown), for the copying model we obtain  $3 \times 10^{-5} \ln^{8.0}(n)$ . Because of the large power of the logarithm required for the polylogarithmic fit, the power-law dependence on  $n$  appears more natural and plausible. (c) and (d) Degree distributions of the two models, histogrammed using adaptive binning (see Appendix C), indicating that  $\gamma_{\text{in}} = 2.1$  and  $\gamma_{\text{out}} = 2.72$ , corresponding to the estimates for the degree distribution of the World Wide Web [13]. In all cases, the mean in and out degree of each network were each two edges per node.

obtain degree parameters other than 3. However, the other two network models can be adjusted to match these values [9, 12]. More details on this are discussed in Appendix A. As before, we set the mean degree to be two in and two out edges per node.

Figure 3 presents the results of these simulations, clearly indicating that  $\delta^{-1}$  scales at least as a power of  $n$ . In particular, we note that the prefactor of the logarithmic fit is over 700 and the power of the logarithm in the polylogarithmic fit is 8, while the power-law fit exponent is close to 1. The results do not change substantially when the mean degree is varied and the degree distributions exponents are fixed. These data indicate that for graphs with degree distributions similar to those measured for the World Wide Web, the GZL adiabatic algorithm for PageRank vector preparation is unlikely to provide an exponential speedup over the classical case.

## V. DISCUSSION

We have investigated the recently proposed adiabatic quantum algorithm for preparing the PageRank vector using an adiabatic quantum algorithm [6]. We find that the eigenvalue gap that determines the algorithm run time depends on the

method of construction of the network, even when the feature believed to be critical for large-scale network structure, the degree distribution, is held fixed. The exponent governing the variation of the gap with graph size does not vary significantly with the method of construction only if power-law scaling of the gap with size is assumed. For networks that are scale-free in their in- and out-degree distributions, and particularly when the degree distributions similar to those measured for the World Wide Web, our numerical results indicate strongly that the GZL adiabatic algorithm for PageRank vector preparation does not offer an exponential speedup over current classical algorithms.

As our research shows that the degree distribution and the average number of edges per node do not fully determine the algorithm's performance, it is not currently known if additional network features contribute to the polylogarithmic gap scaling observed by GZL. Answering this question would be an interesting topic for future work.

## ACKNOWLEDGMENTS

This work was supported in part by ARO, DOD (W911NF-09-1-0439), and NSF (CCR-0635355, DMR 0906951). A.F. acknowledges support from the NSF REU program (PHY-PIF-1104660). We thank S. Garnerone, D. A. Lidar, and D. Bradley for useful discussions. We also thank the HEP, Condor, and CHTC groups at University of Wisconsin-Madison for computational support.

## APPENDIX A: PARAMETERS OF WEB GRAPH MODELS

In implementing the models used in this paper, the relationship between the parameters of the network generation algorithms and the generated networks themselves is not always obvious, so in the following section we explain it in detail.

### 1. GZL preferential attachment

The method of graph construction in the GZL preferential attachment model [6] consists of two phases, each with its own parameter. First, a graph  $X$  (with adjacency matrix  $A_X$ ) is created by adding a new vertex at each time step, where each vertex is created with  $m_X$  outgoing edges. Next, a second graph  $Y$  (with adjacency matrix  $A_Y$ ) is created in the same fashion, only with each new vertex having  $m_Y$  incoming edges.  $A_X$  and  $A_Y$  are then added together, with loops and weights discarded, forming the adjacency matrix of the desired network.  $m_X$  and  $m_Y$  are the two parameters to consider in this algorithm.

In order for a graph to be scale-free,  $\text{Prob}(d_{\text{in}} = k)$  and  $\text{Prob}(d_{\text{out}} = k)$ , the probabilities that the in degree  $d_{\text{in}}$  and the out degree  $d_{\text{out}}$  of a random node have the value  $k$ , must satisfy

$$\begin{aligned} \text{Prob}(d_{\text{in}} = k) &\sim k^{-\gamma_{\text{in}}}, \\ \text{Prob}(d_{\text{out}} = k) &\sim k^{-\gamma_{\text{out}}}, \end{aligned} \quad (\text{A1})$$

where  $\gamma_{\text{in}}$  and  $\gamma_{\text{out}}$  are positive real numbers, and it is understood that  $\text{Prob}(d_{\text{in}} = k) = 0$  when  $k < m_X$  and  $\text{Prob}(d_{\text{out}} = k) = 0$  when  $k < m_Y$ . To compute  $\gamma_{\text{in}}$  and  $\gamma_{\text{out}}$ , one starts from the undirected version from Ref. [18]. This result is then combined with a constant offset, since each vertex of  $X$  has  $m_X$  outgoing edges and each vertex of  $Y$  has  $m_Y$  incoming edges.

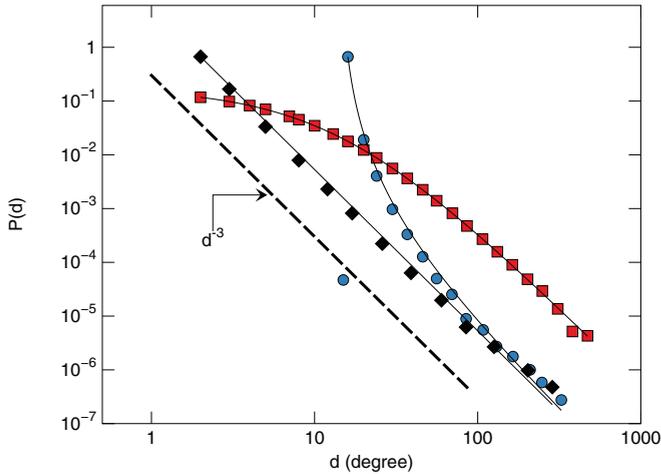


FIG. 4. (Color online) Degree distributions for the GZL preferential attachment model with  $m_X = 1$  and  $m_Y = 15$ , taken at graph size  $n = 8196$  and averaged over approximately 1000 random graph realizations. Both the in-degree (blue circles) and out-degree (red squares) distributions are shown. For reference, the in-degree distribution for  $m_X = 1$  and  $m_Y = 1$  [duplicated from Fig. 2 is shown (black diamonds)]. The dashed line is the expected power law scaling of  $d^{-3}$ , which is applicable for large  $d$ . As predicted by Eq. (A2), shown as fitting curves, the  $m_X = 1$  and  $m_Y = 15$  distributions exhibit non-scale-free behavior over a wide region of  $d$ .

The resulting composite probability distributions follow:

$$\begin{aligned} \text{Prob}(d_{\text{in}} = k) &\sim (k + m_X - m_Y)^{-3}, \\ \text{Prob}(d_{\text{out}} = k) &\sim (k - m_X + m_Y)^{-3}. \end{aligned} \quad (\text{A2})$$

Thus, for sufficiently large  $k$ , these distributions are scale-free. However, for a large range of intermediate  $k$ , we expect substantial deviation from the power-law dependence of Eq. (A1). According to GZL [16], the parameters used to generate Fig. 2 in their paper [6], which provides the main evidence for logarithmic scaling of the gap, follow  $m_Y \gg m_X$ . In Fig. 4 we show the degree distributions for such a network, where we set  $m_X = 1$  and  $m_Y = 15$ . There we see that the degree distributions are well described by Eq. (A2), and that the addition process does indeed distort the degree distributions. By requiring  $m_X = m_Y$ , as we have done in this paper (and GZL did for a portion of their supplemental material [6]),  $\gamma_{\text{in}} = \gamma_{\text{out}} = 3$  for all  $k$ , meaning that the in degrees and out degrees both follow the desired power-law behavior.

The asymptotic (large number of nodes) value of average edges per node for the composite graph is also determined by the parameters  $m_X$  and  $m_Y$ . Because  $m_X$  is the number of outgoing edges per vertex in graph  $X$ , it is also the average number of edges per vertex in  $X$ . The same logic holds for  $m_Y$  and graph  $Y$ . Thus, when constructing the composite graph, the asymptotic average edges per node would be simply  $m_X + m_Y$ . Although loops are then eliminated from the composite graph, the expected number of loops is much less than  $n$  in the large- $n$  case, so this has little effect on the average edges per node. To produce a graph with  $\gamma_{\text{in}} = \gamma_{\text{out}} = 3$  and average in and out edges per node of 2 (as in Fig. 2 of the main text), we use this model with  $m_X = m_Y = 1$ .

## 2. GZL copying model

The parameters of the GZL copying model [6] are similar to the GZL preferential attachment, as they both involve the adding of two graphs to form a composite graph. We again have the parameters  $m_X$  and  $m_Y$ , which again indicate the number of outgoing edges per node in one component graph and the number of incoming edges per node in the other.

This model has two new parameters,  $p_X$  and  $p_Y$ , which are the probabilities of a new node connecting to nodes chosen uniformly at random at a given time step during the construction of  $X$  and  $Y$ , respectively. We follow Ref. [15] and add a constant offset (just as in the preferential attachment case). Doing so, we again obtain the result that the graphs are scale-free only for  $m_X = m_Y$ . Assuming this constraint, the composite graph follows:

$$\gamma_{\text{in}} = \frac{2 - p_X}{1 - p_X}, \quad (\text{A3})$$

$$\gamma_{\text{out}} = \frac{2 - p_Y}{1 - p_Y}. \quad (\text{A4})$$

For the data in Fig. 2 of the main text, we used the parameters  $p_X = p_Y = 0.5$  and  $m_X = m_Y = 1$ . In Fig. 3 of the main text, we used  $p_X = 1/11$  and  $p_Y = 35/86$  and  $m_X = m_Y = 1$ .

## 3. $\alpha$ -Preferential attachment

Just as in the GZL copying model, there are multiple possible actions at each time step in the  $\alpha$ -preferential attachment model [12], and each of these steps has an associated probability.  $p_1$  is the probability of adding a new vertex with a single outgoing edge,  $p_2$  is the probability of adding a new vertex with a single incoming edge, and  $1 - p_1 - p_2$  is the probability of an edge being added to the existing network without the addition of a new vertex.  $\alpha$ , the third parameter, measures how far the generated network deviates from the GZL preferential attachment model.

As laid out in Ref. [12], the relationship between these three parameters and the exponents is

$$\gamma_{\text{in}} = \frac{2 + (p_1 + p_2)\alpha - p_2}{1 - p_2}, \quad (\text{A5})$$

$$\gamma_{\text{out}} = \frac{2 + (p_1 + p_2)\alpha - p_1}{1 - p_1}. \quad (\text{A6})$$

The connection between these parameters and the average number of directed edges per node in the graph is clear when one considers that the probability that a new node will be added at a given time step is  $p_1 + p_2$ , and a new edge is added at each step.

Using these constraints, we can find appropriate values for the parameters for both Figs. 2 and 3 of the main text. In Fig. 2 we used  $p_1 = p_2 = 0.25$ , and  $\alpha = 1$ , and in Fig. 3 we used  $p_1 = 0.415$ ,  $p_2 = 0.0851$ , and  $\alpha = 0.0128$ . These choices in parameters keep  $\gamma_{\text{in}}$  and  $\gamma_{\text{out}}$  fixed at our desired values, while simultaneously keeping the graph at an average of two in and two out edges per node.

## APPENDIX B: INITIAL CONDITIONS

For each of these models, it is necessary to specify an initial graph to seed the network growth. In our simulations we used a complete graph (including loops) with  $m + 1$  vertices, where

$m$  is the number of edges added per vertex (in the  $\alpha$ -preferential attachment model we used  $m = 1$ ).

### APPENDIX C: ADAPTIVE BINNING

In the plots of the degree distributions [Figs. 2(e), 2(f), 3(c), 3(d), and 4], numerical noise caused by few high-degree vertices leads to data which are difficult to interpret. In order

to combat this, we use adaptive binning, which functions as follows. First, some sampling threshold  $s_t$  is set, which we take to be 200 in our analysis. If a given data point, corresponding to a degree, contains at least  $s_t$  samples, then it is included. If the data point instead has fewer than  $s_t$  samples, it is combined with nearby points until the aggregated samples total at least  $s_t$ . The weighted average degree and probability are then recorded.

- 
- [1] P. W. Shor, in *FOCS '94: Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, 1994), pp. 124–134.
- [2] L. K. Grover, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, STOC '96* (ACM, New York, 1996), pp. 212–219.
- [3] D. Bacon and W. van Dam, *Commun. ACM* **53**, 84 (2010).
- [4] S. Brin and L. Page, *Comput. Networks ISDN Syst.* **30**, 107 (1998).
- [5] P. Berkhin, *Internet Math.* **2**, 73 (2005).
- [6] S. Garnerone, P. Zanardi, and D. A. Lidar, *Phys. Rev. Lett.* **108**, 230506 (2012).
- [7] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [8] A. L. Barabasi and R. Albert, *Science* **286**, 509 (1999).
- [9] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády, *Random Struct. Algorithms* **18**, 279 (2001).
- [10] R. Cohen and S. Havlin, *Phys. Rev. Lett.* **90**, 058701 (2003).
- [11] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, in *SIGCOMM '02: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (ACM, New York, 2002), pp. 147–159.
- [12] B. Bollobás, C. Borgs, J. Chayes, and O. Riordan, in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '03* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003), pp. 132–139.
- [13] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, *Comput. Netw.* **33**, 309 (2000).
- [14] F. Chung and L. Lu, *Complex Graphs and Networks (Cbms Regional Conference Series in Mathematics)* (American Mathematical Society, Boston, MA, 2006).
- [15] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, in *Proceedings of the 5th Annual International Conference on Computing and Combinatorics, COCOON'99* (Springer, Berlin, 1999), pp. 1–17.
- [16] S. Garnerone (private communication).
- [17] J. Nelder and R. Mead, *Comput. J.* **7**, 308 (1965).
- [18] A. Barabási, R. Albert, and H. Jeong, *Physica A* **272**, 173 (1999).