

# Measuring Centralization of Online Platforms Through Size and Interconnection of Communities

## Supplementary Material

### S1 Pseudocode Example of Cumulative Disruption Algorithm

For readers seeking a succinct code-like description of our cumulative disruption curve algorithm, we have included listing S1.

```
1 disruption = []
2 for c in communities:
3     remaining = 0
4     original = 0
5     removeCommunity(c)
6     for user in users:
7         if degree(user) > 0:
8             remaining += degree(user)
9             original += originalDegree(user)
10    disruption += [1 - (remaining / original)]
```

Listing S1: Pseudocode for disruption algorithm

Note that when calculating disruption on large networks, it is much more efficient to cache the size of the smallest community that each user participates in. We can then sort all users by the order in which they will be removed, and avoid computationally expensive references to a graph or adjacency matrix for each removal-step in the algorithm.

### S2 Applications to Unipartite Networks

Our influence metric is intended for settings with clearly defined communities. For example, participation in subreddits, membership on a Mastodon server, or committing to a software code repository, all discretely identify users as members of those explicitly-bounded groups. However, network data is often presented in a unipartite configuration such as users following other users. If it is still desirable to delineate communities and measure their influence in these settings, then they can be converted into compatible bipartite networks using the following procedure:

1. Apply a context-appropriate community detection algorithm to label each user as belonging to one community
2. Create a vertex for each community
3. Replace all user-user edges with user-to-community edges, where the edge weight is equal to the number of unipartite edges each user had to other nodes in that community
4. Apply our influence metric to the resulting bipartite graph

An example of this procedure is illustrated in Fig. S1, using a unipartite Watts-Strogatz small-world network (100 nodes, 5 neighbors, rewiring probability of 5%), and label-propagation for community detection. The unipartite graph is shown in the top-left with community labels visualized with color. It is converted to a bipartite representation shown in the upper-right, and the effect of removing each community is illustrated in the bottom frame.

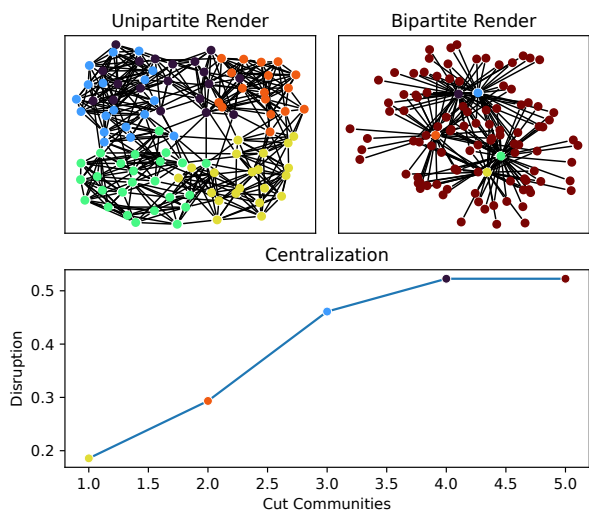


Figure S1: Example of applying our disruption metric to unipartite graphs by detecting communities on a unipartite small-world network (top-left), converting labeled communities into a bipartite representation (top-right), and running our influence metric on the bipartite graph (bottom)

### S3 Calculating the Area Under the Disruption Curve

For Figs. 2b, 3b, and 5 we use the area under the disruption curve as a single-variable summary of how centralized a network is around its largest communities. To calculate the AUC, we use a trapezoidal approximation in logarithmic space.

We chose a trapezoidal approximation to calculate the area even with limited sample points from real-world networks. Integration is possible for purely analytic disruption curve simulations as in Sec. S5, but this is not feasible for our non-Erdős-Rényi networks, so we use a

trapezoidal approximation for all synthetic networks for consistency.

We measure the AUC in logarithmic space, because measuring in linear space would heavily weight the influence of the smallest communities that are removed last, and our primary interest is in examining the influence of the largest communities on the broader population.

## S4 Synthetic Network Topology Details

We measure centralization on a variety of synthetic networks introduced in Sec. 3.3. In this section, we include further description and visualization of the synthetic networks used.

Bipartite Near-Star networks are analogous to a unipartite star network with duplicate edges, but in a bipartite setting. Starting with a unipartite star, replace each edge from the hub to a leaf with a two-path from the hub community to a new “user” vertex, to the leaf community. Duplicate edges from the unipartite hub to leaves are converted into multiple users that share a community, and serve to break ties when pruning communities for disruption curves. This is illustrated in Fig. S2.

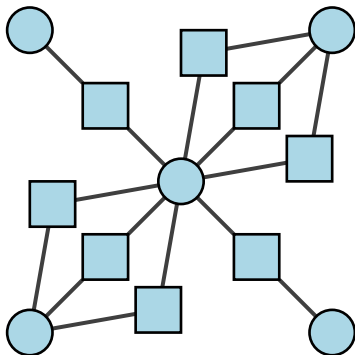


Figure S2: Example Bipartite Near-Star. Circles are communities, squares are users. All users are connected to two communities, and a primary hub is connected to all users.

For our “Powerlaw” networks we follow a bipartite configuration model. We first create vertices representing the desired number of communities and users. We then draw from a powerlaw distribution with an assigned  $\gamma$  exponent, and assign the drawn degree to each community. Then, we create a corresponding number of edges, wiring each community to users drawn uniformly at random without replacement. This yields networks where communities follow a powerlaw degree distribution, while users follow a normal degree distribution.

Bipartite community-user networks can be visualized in a flat plane, as in Fig. 1a, or as a multi-layer graph, as in Fig. S3. A multi-layer representation may be beneficial for representing inter-community relationships that

are not explained by shared users, such as Mastodon federation agreements, or shared moderator staff in two subverses. However, these multiplex relationships were deemed out-of-scope for our current work.

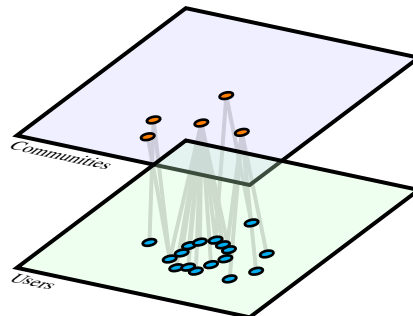


Figure S3: An example bipartite powerlaw network, visualized using a “community” and “user” layer

## S5 Mathematical Analysis of Disruption in Random Networks

We here calculate the disruption curves for random bipartite networks parameterized by their joint-degree distribution. This approach therefore fixes the distribution  $\{g_m\}$  of communities  $m$  per user, the distribution  $\{p_n\}$  of community size  $n$ , and the joint-distribution  $P_{n,m}$  for the degree of the node and community involved in a random bipartite link. Beyond these constraints, the networks are fully random but allow us to explore the role of heterogeneous connectivity at the user and community level as well as the impact of correlations between both levels.

We wish to calculate the disruption  $D(n)$  involved when removing communities of size  $n' < n$  in these random networks. By definition of the bipartite network, we know that  $np_n$  edges are removed when removing communities of size  $n$ . Once again, we define disruption as the fraction of *remaining* edges disrupted by communities of size  $n$  during the pruning process. It is thus given by the number of edges that belong to communities of size  $n$  minus the fraction  $u_n$  of those that are the sole edge of the corresponding users (since these users are removed in the pruning) divided by the number of edges belonging to communities of size equal or smaller than  $n$  minus the  $u_n np_n$  users removed. We write:

$$D(n) = \frac{\overbrace{np_n}^{\text{Edges to comms. of size } n} - \overbrace{u_n np_n}^{\text{Edges to removed users}}}{\underbrace{\sum_{n' \leq n} n' p_{n'}}_{\text{Edges to comms. } n \text{ or smaller}} - u_n np_n} \quad (1)$$

The quantity  $u_n$  can also be defined as the probability that a random user of a community of size  $n$  has no com-

munity smaller than  $n$ . It can therefore be calculated like so:

$$u_n = \sum_m \frac{P_{n,m}}{\sum_{m'} P_{n,m'}} \left( \frac{\sum_{n' \geq n} P_{n',m}}{\sum_{n'} P_{n',m}} \right)^{m-1} \quad (2)$$

Fraction of users in comm. size  $n$  that have  $m$  edges  
Fraction of users with  $m$  edges in comms. larger than size  $n$

In the previous equation, we sum over every possible type of node in a community of size  $n$ , which will have a number of *other* communities  $m - 1$  proportional to  $P_{n,m}$ , and ask for all of these communities to be larger or equal to  $n$ , which will be proportional to the sum of  $P_{n',m}$  over all  $n'$  larger or equal to  $n$ . Normalizing the probabilities appropriately yields Eq. (2) as written.

Note that these equations assume that edges are unweighted, and that there are no duplicate edges, which is what we expect from an infinite random simple graph. In our real-world data sets there are often duplicate edges (for example, one user following several different users on a Mastodon instance), which we compress to weighted edges for convenience.

Despite this difference between the analytical expression and real socio-technical networks, the analysis of random infinite graphs can be useful to test how disruption is impacted by simple network statistics such as degree distributions or correlations in the joint community-user degree matrix  $P_{n,m}$ .

In a simple experiment, we create a random Erdős-Rényi-like bipartite network and correlated equivalent networks with the same degree distributions and variable community-user degree matrices  $P_{n,m}$ . The random network has a simple  $P_{n,m}^{\text{rand}} \propto n p_n m g_m$  (normalized) which we can modify manually. To do so, we calculate the maximally correlated  $P_{n,m}^{\text{max}}$  by assigning users with highest degrees  $m_{\text{max}}$  to the largest communities available before doing the same to users with the next higher degree and so on all the way down. We can do the same to calculate  $P_{n,m}^{\text{min}}$  by assigning users with the lowest degree to the largest communities and working our way up in the user degree distribution. We can then create arbitrary community-user degree matrix  $P_{n,m}$  by interpolating between linearly with  $(1 - \rho)P_{n,m}^{\text{rand}} + \rho P_{n,m}^{\text{max}}$  or  $(1 - \rho)P_{n,m}^{\text{rand}} + \rho P_{n,m}^{\text{min}}$ .

Our results are shown in Fig. S4. We find that positive user-community degree correlations increase disruption and therefore *centralizes* the resulting socio-technical network. Conversely, negative correlations decreases correlations and *decentralizes* the network. That being said, the relative effect of correlations is relatively small as the networks are still otherwise completely random.

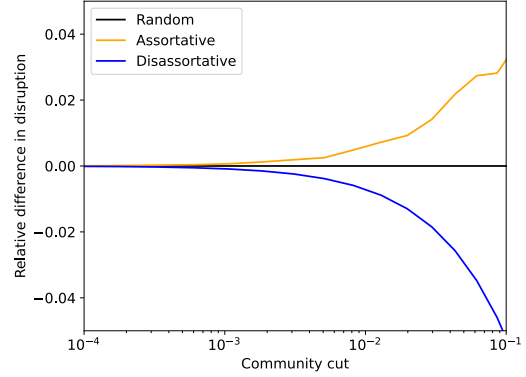


Figure S4: Random bipartite networks with varying user-community degree correlations. We start with a random bipartite network where the ratio of users to communities is 30 and a binomial distribution of communities per user with average 1.2. We create two counterfactuals where degree correlations are at 30% of the maximally assortative network and of the maximally disassortative network. We show the relative difference in disruption caused by correlations (disruption of correlated network divided by disruption of random network minus 1).

## S6 Further Analysis of Assortativity

There are multiple interpretations of degree assortativity in a bipartite setting. The linear correlation between user degrees and community degrees measures whether high-degree users are likely to be connected to high-degree communities. In our network definitions edges represent activity, like follow relationships or participation in conversations, so this measures whether active users are likely to be connected to communities with lots of activity. However, a second metric of interest is whether large communities are likely to be connected to other large communities, or in other words, the assortativity of a unipartite-projected community-community graph. This can also be broken into two sub-cases: assortativity of community size (do communities with many users share users with other high-population communities), and assortativity of degree (do communities with lots of activity share users with other high-activity communities).

These three notions of assortativity are not independent; we might expect that users with lots of activity are active in communities with high populations, and may act as bridges between multiple communities with high activity and high population. However, the three metrics are not guaranteed to correlate and should be measured separately.

While rewiring to promote user-community degree assortativity, we also plotted the changes in community-community degree assortativity, shown in Fig. S5. Strik-

ingly, the community assortativity *decreases* as we rewire to promote user assortativity. This is because as we rewire edges to focus user connections on the largest communities we implicitly decrease the number of edges between communities. This also matches the changes in disruption in Fig. 5: increasing assortativity may reconnect large and insular communities with the rest of the network, briefly increasing their influence, but continued assortativity rewiring also cuts bridges to and between smaller communities, yielding a sparse network that is far less centralized.

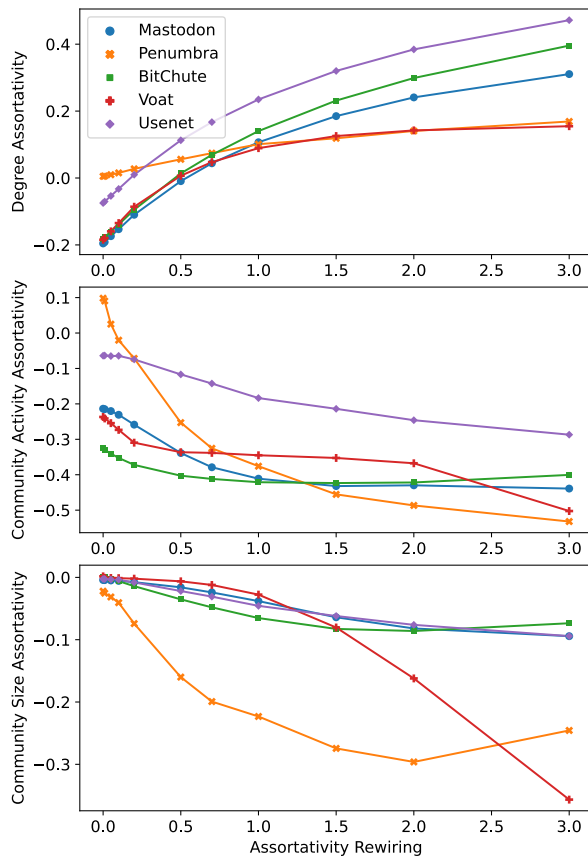


Figure S5: Rewiring to increase user-community degree assortativity (top) decreases the projected community-community degree assortativity (middle) and community-community population assortativity (bottom).

To further explore the relationship between these types of assortativity, we also rewired networks in the reverse direction: for randomly selected pairs of edges, we rewired those edges to *decrease* user to community activity assortativity. We have plotted the change in disruption curves (Fig. S6) and correlation between assortativity metrics (Fig. S7). In most networks, decreasing activity assortativity lowers centralization, although the ef-

fect diminishes as the network topology more closely approximates a random network. The one exception is the Penumbra; this network has such sparse inter-community connections that any perturbation of edges increases the cross-community links and therefore *increases* centralization.

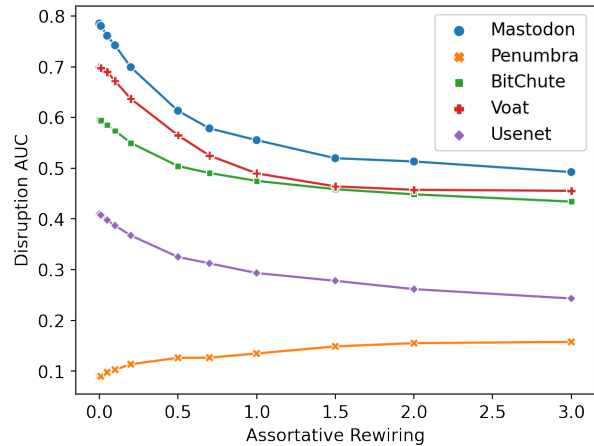


Figure S6: Rewiring networks to decrease user-community degree assortativity also typically decreases disruption when large communities are removed. However, for very sparse networks like the Penumbra, and perturbation, including rewiring to decrease assortativity, increases community inter-connection and so increases the influence of large communities.

## S7 Cumulative Impact on Giant Component Size

Some readers may be interested in how removing large communities influences the giant component size on each network. This is closely related to the cumulative population size in the top sub-plots of Fig. 2a and Fig. 3a. Intuition suggests that the size of the giant component will be inversely proportional to the number of cumulative communities removed; as more large communities are pruned, the giant component should shrink. This relationship holds so long as the remaining communities are interlinked, but falters once a “bridge” community is removed and the giant component splinters. Therefore, sparsely connected networks where bridges are more prominent will have a chaotic giant component size, while more densely connected networks will present a smooth curve until most communities are pruned. This relationship is illustrated in Fig. S8. Most curves are smooth until the tail of the distribution, with two notable exceptions: Voat’s giant component changes once the largest insular communities are removed (see Fig. 4), and the Penumbra’s curve is much “spikier” as a result of its highly sparse structure.

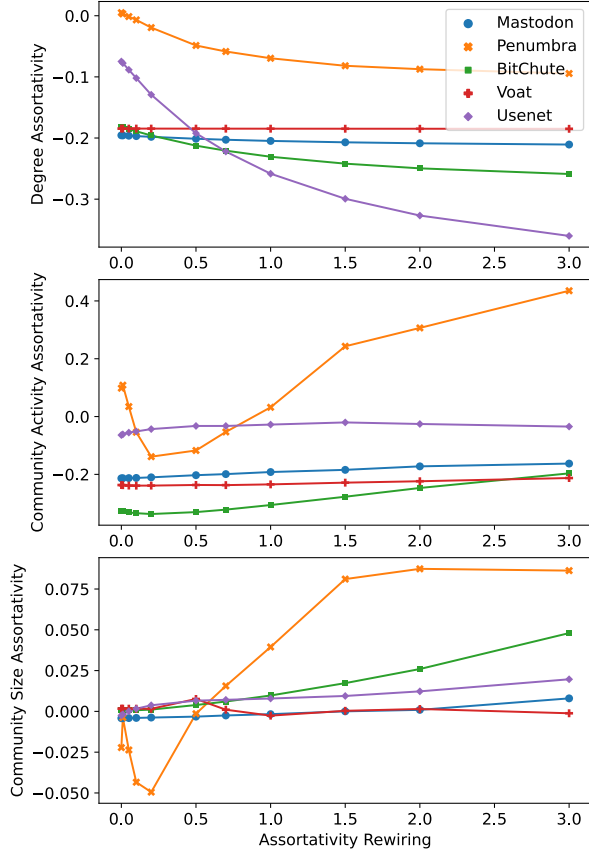


Figure S7: Rewiring to decrease user-community degree assortativity (top) has a small impact on the community-community projected degree assortativity (middle) and projected population assortativity (bottom), except for in the Penumbra; in this sparse network, rewiring first destroys the few active inter-community bridges, then radically increases the interconnectivity of communities.

Measuring the change in giant component size captures some of the same features as our disruption metric. In particular, removing large insular communities may not change the giant component size if the community is completely isolated from the giant component, so this captures some aspect of both the size and topological role of a community. However, the impact of a community is boolean: if it touches the giant component, then removing the community will shrink the giant component by the size of that community. There is no distinction between a minimally integrated and tightly integrated community. Measuring the impact of a community in terms of fraction of edges severed, rather than component vertex size, offers finer insight into the interplay between size distribution and network structure.

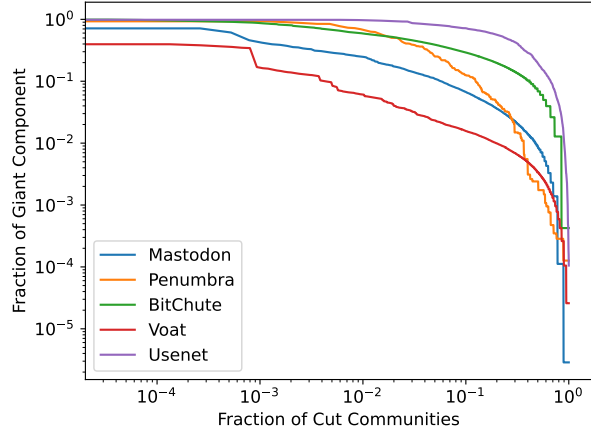


Figure S8: The giant component shrinks as communities are pruned from largest to smallest, indicating both the size of a community and whether it was part of the giant component before pruning. However, this boolean inclusion does not account for how well-integrated the community was among its peers. The y-axis is normalized as a fraction of the un-pruned giant component size, such that “0.5” indicates the giant component is half the size of the original.

### S8 Comparison to Network Bottlenecking

The Cheeger number [1] is a single-valued metric representing how large of a “bottleneck” inhibits conductance across a graph. It is typically written as:

$$h(G) = \min \left\{ \frac{|\partial A|}{|A|} : A \subseteq V(G), 0 < |A| \leq \frac{1}{2} |V(G)| \right\} \quad (3)$$

Edges crossing the boundary of A  
A is a subset of vertices of G  
All edges in+across A  
A contains at most half of all vertices

Our measurement of how much a community influences a larger population, and the Cheeger measurement of whether a community is a “bottleneck” bear some conceptual similarities. Therefore, we compare our metric to the Cheeger number in two ways. First, we create a “local Cheeger number,” following an identical equation  $\frac{|\partial A|}{|A|}$ , but where A is defined as the set of communities we are pruning, rather than via a global search. Second, we estimate bounds on the global Cheeger value of the graph. Since evaluating the graph conductance of all possible subsets of vertices is an NP-hard problem [2], it is impractical to directly measure the Cheeger constant on most large graphs. Fortunately, the Cheeger inequality offers upper and lower bounds on the Cheeger number

based on the second eigenvalue of the normalized Laplacian of the adjacency matrix of  $G$  as follows:

$$\lambda_2/2 \leq h(G) \leq \sqrt{2\lambda_2}$$

Since they are sparse, these bounds can be calculated even on large real-world datasets. Unfortunately, in our tests the bounds are quite wide (see Fig. S9), limiting the utility of this approximation. We have plotted a comparison of the “local” Cheeger number, bounds of the global Cheeger number, and our disruption metric, for a variety of simulated networks.

## References

- [1] B. Mohar, “Isoperimetric numbers of graphs,” *Journal of Combinatorial Theory, Series B*, vol. 47, no. 3, pp. 274–291, Dec. 1, 1989, ISSN: 0095-8956. DOI: 10.1016/0095-8956(89)90029-4. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0095895689900294> (visited on 04/28/2023).
- [2] V. Kaibel, “On the expansion of graphs of 0/1-polytopes,” in *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, SIAM, 2004, pp. 199–216.

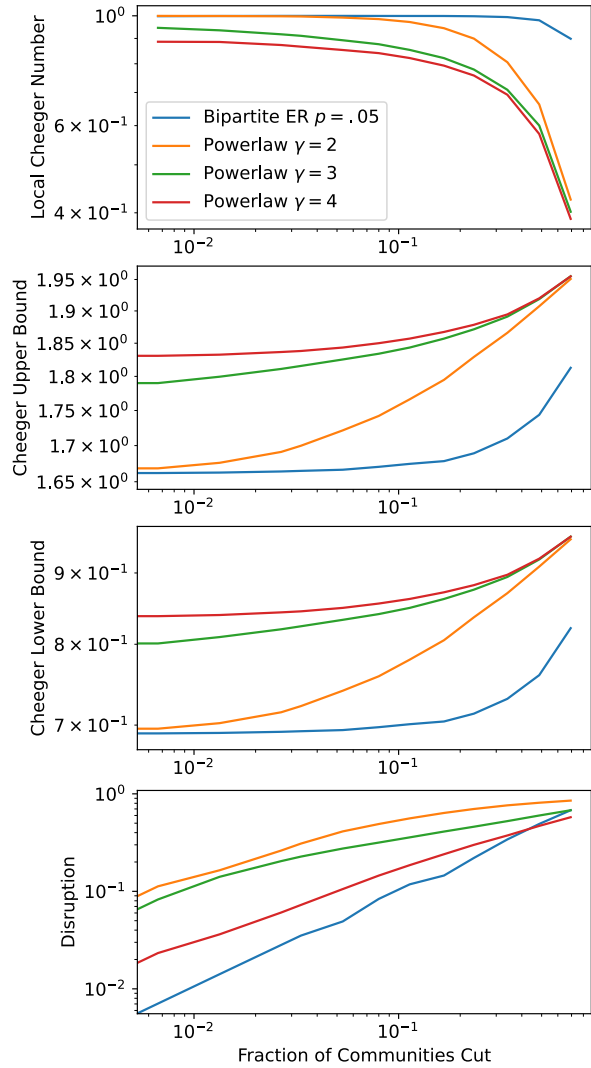


Figure S9: Our network disruption metric bears some conceptual similarity to network bottlenecks, but neither a “local” Cheeger value measuring the bottleneck effect of removed communities (top) nor upper- and lower-bound estimates of the global Cheeger number describe the same trends.