Abstract—Random graph models are important constructs for data analytic applications as well as pure mathematical developments, as they provide capabilities for network synthesis and principled analysis. Several models have been developed with the aim of faithfully preserving important graph metrics and substructures. With the goal of capturing degree distribution, clustering coefficient, and communities in a single random graph model, we propose a new model to address shortcomings in a progression of network modeling capabilities. The Block Two-Level Erdős-Rényi (BTER) model of Seshadhri et al., designed to allow prescription of expected degree and clustering coefficient distributions, neglects community modeling, while the Generalized BTER (GBTER) model of Bridges et al., designed to add community modeling capabilities to BTER, struggles to faithfully represent all three characteristics simultaneously. In this work, we fit BTER and two GBTER configurations to several real-world networks and compare the results with that of our new model, the Extended GBTER (EGBTER) model. Our results support that EGBTER adds a community-modeling flexibility to BTER, while retaining a satisfactory level of accuracy in terms of degree and clustering coefficient. Our insights and empirical testing of previous models as well as the new model are novel contributions to the literature.

I. INTRODUCTION

Graphs provide natural representations of objects and their relationships, and are now a prevalent tool in the applied sciences, including biology, physics, and computer science [1–12]. Random graph models, which prescribe probability distributions over a set of graphs, have become a mainstay for both applied and pure graph theory [13–20]. They give probabilistic machinery admitting principled network simulation capabilities that facilitate statistical analysis in the face of data that is scarce, time-consuming, or impractical to apprehend. Complex networks, especially those arising from natural representations of real world data, exhibit structural idiosyncrasies, which are measured and deduced using many heterogeneous graph metrics, e.g., distributions of degrees, clustering coefficients, and motifs as well as measures of size, centrality, and modularity, to name a few. Consequently, a major focus in random graph models has been on identifying those graph metrics that are important in real-world graph representations and on developing models that preserve these metrics [14, 17, 18, 21–30].

This paper identifies and addresses gaps in a progression of graph models presented recently in the research literature. Building on the famous Erdős-Rényi (ER) and Chung-Lu (CL) models, the Block Two-Level Erdős-Rényi (BTER) introduced by Seshadhri et al. (2012) [17] was designed to allow specification of degree distribution and clustering coefficient per degree. For understanding social networks, clustering coefficient is an important metric as it is driven by the nature of real-world relationships. Initial tests show accurate modeling of clustering coefficients per degree as well as the degree distribution by BTER. Further work of Kolda et al. [25] discuss scalable implementations of BTER.

Similarly, understanding community structure (number, size, within- and between-community densities) and membership (who i.e., which nodes, interacts with whom) is important for network analysis, as it shows important relationships and their changes. As BTER does not allow specification of communities, Bridges et al. (2015) introduced a Generalized BTER model, (GBTER) [19, 20] adding user flexibility to prescribe community membership and community edge densities, while retaining the flexibility to also prescribe (most) degree distributions in expectation. As initial efforts with GBTER focused on aiding anomaly detection algorithms (specifically, GBTER’s probabilistic formulation allowed p-value computations, which were used for identifying anomalous changes in node degree and community membership in time-varying graphs), GBTER’s efficacy in modeling communities while preserving degree or clustering coefficient properties is unknown.

In this work we investigate how well GBTER configurations model many real world networks. We identify shortcomings of this model and provide quantitative results and explanations for why the model fails to preserve desired characteristics. In light of these findings, we introduce an improved model, the Extended GBTER (EGBTER), and exhibit it’s modeling capabilities on the same set of networks. An advantage of the GBTER and EGBTER contributions (as well as Stochastic Block Model [16]) is that one can model not just community structure (in terms of number of communities, size and density...
of each, and interactions between communities) but also specify membership.

Our contributions provide qualitative comparisons of the BTER model and two configurations of the GBTER model, as well as quantifiable head-to-head results by using three graph metrics that reflect edge, clustering, and community structure statistics on several real-world networks. Our testing gives insight to the modeling capabilities of these relatively new models, and, broadly speaking, shows that BTER often succeeds in faithfully reproducing clustering metrics at the expense of community structure, while GBTER performs conversely.

Informed by our analysis of BTER and GBTER, we introduce the EGBTER model, a natural combination of BTER and GBTER, addressing the limitations of both. More specifically, EGBTER is crafted to circumvent degree distribution problems within user-defined communities that are caused by the simplicity of the ER process used in the GBTER model, but retains the ability to specify community membership and density. We compare EGBTER both qualitatively and quantitatively to the previous two models. Our results suggest that EGBTER provides more balanced modeling of degree distribution, clustering coefficient, and community structure. We note that many other random graph models seek to preserve similar characteristics. E.g., the Stochastic Block Model [16] seeks to faithfully preserve density between specified nodes. The dK-Orbis models [18] prescribe distribution of edges between d-tuples of node degrees and have exhibited preservation of community structure and degree distribution for large enough d (but do not allow specification of which nodes constitute each community). Comparison of EGBTER against other models besides BTER and GBTER is necessary for greater understanding, but is outside of scope for this work.

II. Notation & Previous Models

We use classical terms and notation except in the case of clustering coefficient, where we use average local clustering coefficient in place of clustering coefficient of a node. We shall deal exclusively with simple graphs. See Table I for notation and definitions used throughout. We use density to refer to the percentage of total possible edges present in a network. A triangle refers to a complete graph on three nodes. In the literature, notions of community and community structure vary. In this work, we use the algorithm of Louvain et al. [31] to produce a partition of $V$. We refer to each set of nodes in this partition as a community and the partition itself as the community structure.

Next, we describe the previous models, BTER and GBTER. Both are built on combinations of two influential and historical models, ER and CL, which are described in the Appendix.

**BTER Model:** The BTER model uses a two-step edge insertion process. In the first stage, nodes are partitioned based on the given expected degree sequence, and an ER model is sampled in the subgroups with parameter chosen to preserve the clustering coefficient per degree (CCPD). Next a CL model is applied to ensure the expected degree sequence is attained.

**Inputs:**
1. Expected degree distribution \( \{d\} \)
2. Expected CCPD \( \{cc_d\} \)

**Generative process:**
1. Node groupings are implicitly assigned by putting nodes of degree \( d \) into groups of size \( d + 1 \) and denoted by \( A_d \).
2. Build an ER graph in each \( A_d \) of size \( d + 1 \) nodes with \( p_{A_d} = cc_d^{1/3} \).
3. Build a CL graph on the entire network with node weights \( w_i := \max\{0, d_i - cc_d^{1/3}(|C_k| - 1)\} \).

![Fig. 1: BTER process depicted. (Left to right) nodes partitioned into groupings \( A_d \), within-group edges generated from ER process, edges from CL process generated. Figure from Seshadhri et al. [17].](image-url)
model, we propose the EGBTER model. Unlike the previous membership and density of communities in a single graph and Seshadhri et al. [17, 25]. For a more detailed description, see the works of Kolda et al. See Fig. 1 for a visual depiction of the edge generation process.

Generative process:
1) Each edge is added with probability
2) For each within-community grouping, store
3) Expected density parameter, \( p_k \) for each community \( C_k \) (modeling within-community density or within-community CCPD)

\[
P(\{v_i, v_j \in E\}) = \begin{cases} 
  p_k + (1 - p_k)CL(\epsilon_i, \epsilon_j), & \text{if } v_i, v_j \in C_k \\
  CL(\epsilon_i, \epsilon_j), & \text{otherwise},
\end{cases}
\]

where \( \epsilon_i = \max\{0, d_i - p_L(D)\} \) and \( E_i = D_i - d_i \), respectively. Community density, \( p_L \), is the ER probability, and \( D \) is the number of neighbors of \( v_i \) in this grouping.

4) Within each community \( C_k \) run a BTER process—ER with parameter \( p_k \) and CL with node weight \( \epsilon_i \) for all nodes \( i \in C_k \).
5) For all nodes \( v_i \in V \), run global CL process with node weight \( E_i \).

IV. Experiment

We compare the BTER, GBTER and EGBTER models on faithful preservation of degree distribution, CCPD distribution community structure for seven real-world networks. For degree distribution and CCPD evaluation, the root mean squared error is computed against the original graph’s corresponding feature sequence. For community structure, we compare modularity and present a few telling visualizations of real and simulated graphs. Modularity is a graph metric taking values in \([-0.5, 1]\] with higher scores indicating stronger community structure in two models, which use ER generation within groupings of nodes (to fulfill CCPD or community density expectations) and CL generation across all nodes (to fulfill the expected degree distribution), the EGBTER model exchanges the within-community ER process (used in the GBTER model) for a BTER analogue. The motivation for doing so is to achieve greater overall accuracy by addressing the deficiencies that may occur at the within-community stage. While GBTER allows modeling of dense sub-regions (communities), the ER processes poorly model within-community degree distributions, as well as clustering coefficients; for example, see Figs. 2, 3, 4. The critical insight is that (a) degree distribution and CCPD is important in modeling graphs and these metrics are heavily influenced by edge generation within tight-knit communities, (b) neither are modeled well by ER (and therefore not by GBTER), but (c) both are modeled well by BTER at the graph level; hence, we hypothesize that using a BTER process inside each specified community (and CL process between communities) will give better overall modeling of degrees, CCPD, and community membership and density. Explicit details are given below.

\[
P(\{v_i, v_j \in E\}) = \begin{cases} 
  p_k + (1 - p_k)CL(\epsilon_i, \epsilon_j), & \text{if } v_i, v_j \in C_k \\
  CL(\epsilon_i, \epsilon_j), & \text{otherwise},
\end{cases}
\]

where \( \epsilon_i = \max\{0, d_i - p_L(D)\} \) and \( E_i = D_i - d_i \), respectively. Community density, \( p_L \), is the ER probability, and \( D \) is the number of neighbors of \( v_i \) in this grouping.

4) Within each community \( C_k \) run a BTER process—ER with parameter \( p_k \) and CL with node weight \( \epsilon_i \) for all nodes \( i \in C_k \).
5) For all nodes \( v_i \in V \), run global CL process with node weight \( E_i \).

IV. Experiment

We compare the BTER, GBTER and EGBTER models on faithful preservation of degree distribution, CCPD distribution community structure for seven real-world networks. For degree distribution and CCPD evaluation, the root mean squared error is computed against the original graph’s corresponding feature sequence. For community structure, we compare modularity and present a few telling visualizations of real and simulated graphs. Modularity is a graph metric taking values in \([-0.5, 1]\] with higher scores indicating stronger community structure in two models, which use ER generation within groupings of nodes (to fulfill CCPD or community density expectations) and CL generation across all nodes (to fulfill the expected degree distribution), the EGBTER model exchanges the within-community ER process (used in the GBTER model) for a BTER analogue. The motivation for doing so is to achieve greater overall accuracy by addressing the deficiencies that may occur at the within-community stage. While GBTER allows modeling of dense sub-regions (communities), the ER processes poorly model within-community degree distributions, as well as clustering coefficients; for example, see Figs. 2, 3, 4. The critical insight is that (a) degree distribution and CCPD is important in modeling graphs and these metrics are heavily influenced by edge generation within tight-knit communities, (b) neither are modeled well by ER (and therefore not by GBTER), but (c) both are modeled well by BTER at the graph level; hence, we hypothesize that using a BTER process inside each specified community (and CL process between communities) will give better overall modeling of degrees, CCPD, and community membership and density. Explicit details are given below.

\[
P(\{v_i, v_j \in E\}) = \begin{cases} 
  p_k + (1 - p_k)CL(\epsilon_i, \epsilon_j), & \text{if } v_i, v_j \in C_k \\
  CL(\epsilon_i, \epsilon_j), & \text{otherwise},
\end{cases}
\]

where \( \epsilon_i = \max\{0, d_i - p_L(D)\} \) and \( E_i = D_i - d_i \), respectively. Community density, \( p_L \), is the ER probability, and \( D \) is the number of neighbors of \( v_i \) in this grouping.

4) Within each community \( C_k \) run a BTER process—ER with parameter \( p_k \) and CL with node weight \( \epsilon_i \) for all nodes \( i \in C_k \).
5) For all nodes \( v_i \in V \), run global CL process with node weight \( E_i \).
relation to the given partition; i.e., high modularity indicates high density of edges within communities and sparse edges across communities. Borrowing vocabulary from the clustering literature \[33\] modularity is internal validation, i.e., unsupervised, metric. Modularity is computed after designating the community structure of the simulated network using the algorithm of Louvain et al. \[31\]. See Table I for metric formulas. For each metric, we macro-averaged results across 100 simulations for each model using the same input parameters.

**Real-world networks description:** All networks used in this work for model evaluation can be found online either from the Network Repository \[34\] or from the Stanford Network Analysis Project (SNAP) \[35\]. Two networks come from biology (a protein interaction network \[36\] and a fly brain network), two are social networks (a Facebook friend network \[32\] and physics collaboration network \[37\]), two are web networks \[38-40\] and one is an infrastructure network \[15\]. Only the two social networks come from SNAP.

**BTER model generation:** For the BTER model, we use the online Matlab implementation \[41\] provided by BTER authors, Kolda et al. \[17, 25, 28\]. Step-by-step instructions for fitting BTER to real-world networks are provided online \[41\]. The supplied code measures/estimates degree and CCPD distributions before sampling edges for the ER and CL processes. We use the default value of any additional parameters. The edge list of each network was exported to Python where graph metrics were computed using the Networkx package \[42\].

GBTER model generation: By design, GBTER admits ample flexibility of configuration, in particular, in defining communities members and prescribing within-community ER probabilities. Bridges et al. configured the model by learning \(p_k\) as within-community density of community \(C_k\), but also mentioned that within-community CC could be modeled by setting \(p_k = cC_k^{1/3}\). We configure GBTER with each of the above two possibilities. To partition vertices into communities (for GBTER and EGBTER fitting), community assignments are determined by using the Louvain community detection algorithm \[31\] in Python \[43\].

For each real-world network, a Networkx graph object \(G\) is created. We find \(\{C_k\}\) with Louvain’s algorithm, and for each \(C_k\) we measure/compute density and average local CC on \(G[C_k]\). We also computed \(\epsilon_i = \max\{0, d_i - p_k(|C_k| - 1)\}\) for each node \(v_i\). For edge insertion, we iterate over all possible edges and add the edge with probability given by Eqn. 1.

EGBERT model generation: To generate an EGBTER network, we measure inputs from the original network, and for each \(C_k\) we generate and store the BTER community groupings, \(A_L\), using global degrees, \(D_i\), to determine \(A_L\). In each \(A_L\) we use \(p_L = \sqrt{CCD}\) of the node with lowest within-community degree \(\{d_i\}\) in \(A_L\) to compute \(\{\epsilon_i\}\) (Refer to Sec III). We then sample edges as done in the scalable BTER model implementation \[25\].

For each edge, we sample from a trinomial distribution to determine which process generated the edge (ER, within-\(C_k\) CL, or global CL process). Weights for each CL process are given by the sum of the excess expected degree sequence divided by total weight for all three processes. In the ER process, sampling gives rise to duplicates so in each grouping \(A_L\), if we wish to add \((\frac{|A_L|}{2})\) distinct edges, we must perform \(w(A_L) = (\frac{|A_L|}{2})\log\{(1 - p_k)^{-1}\}\) samples in expectation. See the work of Kolda et al. \[25\] for a proof. The ER process total weight is \(\sum A_L w(A_L)\) divided by total weight across all three processes.

Once we determine which process generated the edge, we proceed to sample edge endpoints according to the corresponding edge criteria for that process. We repeat the weighted calculation above but on the level of \(A_L, C_k\) or globally depending on predetermined edge process. For example, suppose in the first step we determine that our edge is generated in some \(ER\) process. Next we sample \(A_L\) (a group of nodes) with \(P(A_L) \propto w(A_L)\). Once \(A_L\) is determined, we sample two endpoints, without replacement, uniformly from \(\{v_i \in A_L\}\)

### Table II: Color hierarchy (best to worst) is blue, light blue, light red, red, for each network & metric. Across all networks, BTER is usually first or second for degree & CC distributions, yet usually poor in modularity; EGBTER is usually best or second in all three metrics. All metrics are macro-averaged across 100 random realizations for each model.

| Model       | RMSE\([d]\) | \(Q\) | RMSE\([ccd]\) |
|-------------|-------------|------|--------------|
| bio-dmela   | true        | NA   | 0.4530       | NA            |
|             | bter        | 59.3526 | 0.3445   | 0.0113        |
|             | gbter       | 112.1429 | 0.4335   | 0.0105        |
|             | gbter\[CC\] | 127.3613 | 0.7087   | 0.0118        |
|             | egbter      | 61.3373  | 0.4620    | 0.0181        |
| facebook    | true        | NA   | 0.8350       | NA            |
|             | bter        | 2.4670  | 0.7160     | 0.0973        |
|             | gbter\[CC\] | 10.2906 | 0.0648    | 0.2021        |
|             | egbter      | 9.7372  | 0.8728     | 0.0768        |
|             | gbter\[CC\] | 13.9274 | 0.1948    | 0.1748        |
| hub-fly     | true        | NA   | 0.4188       | NA            |
|             | bter        | 5.17292 | 0.5565    | 0.0958        |
|             | gbter\[CC\] | 15.7820 | 0.4997    | 0.0464        |
|             | egbter      | 18.4049 | 0.7520    | 0.0799        |
|             | gbter\[CC\] | 7.4551  | 0.5149     | 0.0307        |
| ca-GQe      | true        | NA   | 0.8628       | NA            |
|             | bter        | 75.9245 | 0.8093    | 0.3202        |
|             | gbter\[CC\] | 84.0651 | 0.6567    | 0.4004        |
|             | egbter      | 111.2112 | 0.9237   | 0.4688        |
|             | gbter\[CC\] | 82.4652 | 0.8617    | 0.3476        |
| mil-email   | true        | NA   | 0.9357       | NA            |
|             | bter        | 151.1312 | 0.7322    | 0.0461        |
|             | gbter\[CC\] | 184.9929 | 0.7711    | 0.0755        |
|             | egbter      | 236.2863 | 0.9266   | 0.1900        |
|             | gbter\[CC\] | 190.5787 | 0.9399    | 0.0346        |
| webbase-2001| true       | NA   | 0.5002       | NA            |
|             | bter        | 21.3190 | 0.5310     | 0.0590        |
|             | gbter\[CC\] | 48.3568 | 0.4529    | 0.0984        |
|             | egbter      | 57.91125 | 0.6568   | 0.2235        |
|             | gbter\[CC\] | 29.1909 | 0.5693    | 0.0566        |
|             | gbter\[CC\] | 126.9480 | 0.7437    | 0.0494        |
|             | gbter\[CC\] | 178.8203 | 0.7868    | 0.0600        |
|             | egbter      | 225.2118 | 0.4467    | 0.2136        |
|             | egbter      | 134.7270 | 0.9367    | 0.0340        |
Comparison of web-spam [38], BTER and EGBTER degree distributions.

Comparison of web-spam and GBTER models degree distributions.

Fig. 3: Note the similar fit of the BTER and EGBTER model in matching the degree distribution for this particular network. GBTER (CC) model’s degree distribution is dominated by within-community ER models causing unrealistic fit.

V. RESULTS

Table II summarizes the numerical output of our simulations.

Degree Distribution Results: We see that the BTER model performs the best in $RMSE\{d\}$ across all seven networks and the EGBTER model performs second best in six of seven networks. Configuring GBTER to model within-community CC in each community produces the worst error across all networks. This is because for the networks used, within-community CC exceeded within-community density, causing the model to overstate the number of edges in each community in the ER process. When this happens, the local surplus in each community translates to a global degree distribution that is a superposition of the overstated ER degree distributions from each of the communities. This arises from the implicit assumption in GBTER that communities are well-modeled internally by an ER model. See Fig. 2 and Fig. 3b for visualizations of GBTER degree distributions when within-community CC exceeds within-community density.

Clustering Coefficient per Degree Distribution Results: EGBTER does the best in four of seven networks on minimizing $RMSE\{cc_d\}$ (global CCPD distribution). This is to be expected as we incorporate an analog of the BTER process on the community level to adhere to the within-community CCPD distribution. Despite incorporating community structure into our model, we gain a slight advantage in $\{cc_d\}$. This is not quite clear from Fig. 4a, however is evident from the numerical computation (Table II).

The deficiency of GBTER in modeling $\{cc_d\}$ can be seen in Fig. 4b. The CCPD of high degree nodes is grossly overstated. As with the previously observed edge surplus (Fig. 3b), we attribute this to the ER process used within communities in the GBTER model.

Community Structure Results:

On modularity, EGBTER performed the best in five of seven networks with GBTER (modeling within-community density) coming in second. This indicates that EGBTER does better in preserving the overall level of community structure in the network than the other models.

Note that when modeling within-community CC, GBTER tends to overstate the overall level of network division while modeling within-community density tends to have the opposite effect. This agrees with previous discussion on edge metrics where we saw that the degree distribution was a superposition of ER degree distributions from each community when modeling within-community CC. See Fig. 5 for network visualizations.

The BTER model does not perform as well as EGBTER and GBTER in preserving modularity, only performing better than the GBTER model when modeling within-community CC. This is unsurprising as it does not allow user prescription of community structure. Visualizations (using the software Gephi [44]) of the outputs generated by the BTER model tend to exhibit an organized star-shaped collection of dense ER graphs together with, in some cases, a large dense core (see Fig. 5c and Fig. 5f). The visualization BTER produced depended heavily on which subprocess generated the majority of edges in the underlying simulated network. The star-shaped collection arose when the ER process was the dominant edge generation process, while the dense core arose when the CL model was the dominant edge generation process. Each distinct type of visualization agrees with previous work on both the BTER model [17] and on average distances in CL graphs [14, 45].

VI. CONCLUSION

The EGBTER model presented here showed strong overall performance across metrics measuring degree distribution, clustering coefficients, and community structure. The BTER model uses an automated grouping process that does well to
(a) Comparison of web-spam [38], BTER and EGBTER CCPD distributions.

(b) Comparison of web-spam and GBTER models’ CCPD distributions.

Fig. 4: Visual depiction of web-spam [38] degree distribution vs. simulated analogs.

match both degree and CCPD distribution but fails to preserve community structure at the coarsest scale. This problem is addressed in the GBTER model which imposes community membership and size. However, this comes at a cost to CCPD, as was found to be the case in both configurations of GBTER tested here.

This work was concerned with simultaneous faithful preservation of particular graph characteristics. Performance and scalability are out of the scope of our consideration. Testing showed that in networks with high modularity, internal and global degree and clustering coefficients were nearly identical, except for high degree nodes. Recent work [28, 46] has shown CCPD distribution can be estimated via sampling and scales to large networks. Future research is needed to uncover explicit relationships and network properties that would allow for autonomous and scalable generation of EGBTER graphs without the direct reliance on the measured inputs from a seed graph. Overall, we hope our contributions will enhance the modeling and analysis capabilities graph theory brings to the many diverse applications.

ACKNOWLEDGEMENTS

This research was supported in part by the Integrated Joint Cybersecurity Coordination Center (iJC3) Cyber R&D program, of the U.S. Department of Energy. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor of this work.

APPENDIX

Here we describe the metrics used to describe how well different graph models preserve important network characteristics. The three metrics we use are root mean squared error (RMSE) of both degree distribution and CCPD distribution, as well as modularity as a measure of community structure.

Degree Distribution: To measure how well the degree distribution of a simulated network fits the degree distribution of the original network, we use the root mean squared error (RMSE) applied to the simulated degree distribution using the degree distribution of the original network as a baseline.

CCPD Distribution: We apply the RMSE of the CCPD distribution in the same fashion as above. This measure implicitly depends on the set of all degrees produced in the simulated network as we measure the error from the set of ordered pairs \(\{(d, cc_d)\}\) in the simulated network to the original network.

Modularity: To evaluate how well a particular vertex clustering (or in our case a partition) captures the community structure of a network, we shall primarily use the modularity metric formulated by Newman [47]. Modularity, denoted \(Q\), is given by,

\[
Q = \sum_i e_{ii} - a_i^2,
\]

where \(e_{ij}\) is the fraction of total edges that connect nodes in community \(C_i\) to community \(C_j\) and \(a_i = \sum_{j} e_{ij}\). We use this metric because it considers the vertex partition as a whole and allows us to compare different partitions of the same network. Modularity gives a sense of the overall level of community structure present in a network relative to the given partition. This is not to say that higher modularity means better community structure but that higher modularity indicates more overall division into communities with respect to the partition used.

ER Model: The Erdős-Rényi (ER) model [48] was one of the first random graph models. The ER model takes as input, the desired number of nodes \(n\) and an edge insertion probability \(p\). Then a graph on \(n\) nodes is generated by inserting each of the \(\binom{n}{2}\) possible edges with probability \(p\). This graph is denoted as \(ER(n, p)\). As a result of this simple insertion process, in expectation we have network density, \(cc\), and CCPD equal to \(p\). We also have \(\binom{n}{2}p\) edges in expectation and

\[
P(d_i = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}.
\]

This produces a roughly Gaussian shaped degree distribution, counter to that observed in real world networks [22].
Fig. 5: Visualizations of FB network and simulated networks according to model. Following quantitative modularity results, EGBTER qualitatively simulates community structure best. We also include a BTER visualization of a biological protein interaction network (f), as BTER tended to exhibit either a star pattern (c) or a dense core (f). All visualizations were produced in Gephi [44].

**CL Model:** The Chung-Lu (CL) model [49] arose as an attempt to model graphs with a power law degree distribution. In the CL model, each node $v_i$ is assigned a weight, $w_i$. The CL model takes as input, the list of corresponding vertex weights $\{w_i\}$. The probability of inserting edge $v_i v_j$ is

$$P(v_i v_j \in E) = \frac{w_i w_j}{\sum_{v_k \in V} w_k},$$

which we shall denote by $CL(w_i, w_j)$. For this probability to be well defined we require that $w_i w_j \leq \sum_{v_k \in V} w_k$ for all $v_i, v_j \in V$. It is possible that the RHS of Eqn. 3 is greater than 1. For our purposes, if this occurs we set $P(v_i v_j \in E) = 1$. A common adaptation is to use $w_i = d_i$ for all $v_i \in V$. When $w_i = d_i$, this is referred to as a null model. CL generalizes ER and retains independent edge insertion while being able to match $d_i$ in expectation. However, this process rarely closes triangles [25] and the average distance is relatively small [14, 45]. This makes this model a bad fit for CC and community structure but laid a foundation for capturing network degree distribution in expectation. See Fig. 5f for BTER visualization of a biological network. This visualization looks very similar to those of CL models.

**REFERENCES**

[1] T. Aittokallio and B. Schwikowski, “Graph-based methods for analysing networks in cell biology,” Briefings in Bioinformatics, vol. 7, no. 3, p. 243, 2006. [Online]. Available: http://dx.doi.org/10.1093/bib/bbl022

[2] A. Goldenberg, A. X. Zheng, S. E. Fienberg, E. M. Airoldi et al., “A survey of statistical network models,” Foundations and Trends® in Machine Learning, vol. 2, no. 2, pp. 129–233, 2010.

[3] S. Pirzada, “Applications of graph theory,” PAMM, vol. 7, no. 1, pp. 2070013–2070013, 2007.

[4] N. Pržulj and T. Milenkovi, “Computational methods for analyzing and modeling biological networks,” Biological Data Mining. Chapman & Hall/CRC, pp. 397–428, 2009.

[5] D. Karlan, M. Mobius, T. Rosenblat, and A. Szeidl, “Trust and social collateral,” The Quarterly Journal of Economics, vol. 124, no. 3, pp. 1307–1361, 2009.

[6] J. Liu, E. Bier, A. Wilson, T. Honda, K. Sricharan, L. Gilpin, J. A. G. Gómez, and D. Davies, “Graph analysis for detecting fraud, waste, and abuse in healthcare data.” in AAAI, 2015, pp. 3912–3919.
“Counting triangles in massive graphs with mapreduce,” SIAM Journal on Scientific Computing, vol. 36, no. 5, pp. S48–S77, 2014.

[47] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” Physical review E, vol. 69, no. 2, p. 026113, 2004.

[48] P. Erdős and A. Rényi, “On random graphs,” Publicationes Mathematicae Debrecen, vol. 6, pp. 290–297, 1959.

[49] W. Aiello, F. Chung, and L. Lu, “A random graph model for power law graphs,” Experimental Mathematics, vol. 10, no. 1, pp. 53–66, 2001.