# Appendix: Supplementary Material for Algorithmic Tools for Understanding the Motif Structure of Networks

No Author Given

No Institute Given

## A Section 3 - CombArt Algorithm

Here we provide the pseudocode of GreedyPeeling used in the CombArt Algorithm.

**Algorithm 1:** GreedyPeeling($G(V, E), \mathcal{M}$)

```
Initialize $V_0 = V, i = 1$;
For each node $v \in V$, count the number of motif $\mathcal{M}$ it participate in as
  $\deg_{\mathcal{M}}(v)$;
while $i < |V|$ do
    $u \leftarrow \arg\min_{u \in V_{i-1}} \deg_{\mathcal{M}}(u)$ (break tie arbitrarily);
    $V_i = V_{i-1} \backslash u$;
    For each node $v \in V_i$ update $\deg_{\mathcal{M}}(v)$ according to $G[V_i]$;
    $i \leftarrow i + 1$;
return $V_j \in \{V_0, \ldots V_{|V|-1}\}$ such that maximizes $\frac{m(V_j)}{|V_j|}$. ;
```

## B Section 5.2 - Bitcoin trust networks

Figures 1(a),(b) visualize two anomalous subgraphs found in Bitcoin-OTC and Bitcoin-Alpha respectively. In the Bitcoin-OTC trust network we contrast the motif pair $(11, 9)$, while in the Bitcoin-Alpha trust network the motif pair $(11, 4)$. The motifs and their IDs are shown in Figure 2 in the main text. The choice of the pairs was done according to the same methodology described in Section 5.2, p.11, and is based on the $z$-scores, see Figure 1(f). In Figure 1(a) we detect a set of nodes that is likely to be fraudsters/scammers; they vote for each other positively, but the rest of the nodes in the graph gives them negative ratings. The fraudster nodes are shown in a box (one node), and in the bottom half of the visualization as indicated by the black line. Distrust edges are shown in red, while trust edges are shown in green.

A similar pattern is found in Bitcoin-Alpha as shown in Figure 1(b). We observe that two new accounts have been opened with only positive ratings from a set of fraudsters/scammers who have been distrusted by normal nodes in the community. This behavior matches a well-known scheme where scammers create multiple accounts to increase their own ratings and to reduce the ratings of benign users, see [12]. Finally, Figures 1(c), (d) show the histogram of ratings in the two detected subgraphs. The global histogram of ratings is similar between the two graphs; the aggregated histogram of ratings is shown in Figure 1(e). The difference between histograms in Figures 1(c), (d) and Figure 1(e) indicate that the detected subgraphs are anomalous.

## C Section 6, Motif Significance and Null Models

Before we describe our findings in Section C.2 for Q3 (as posed in Section 6 of the main text), we provide an overview of random graph models, including the seven models we use, in Section C.1.

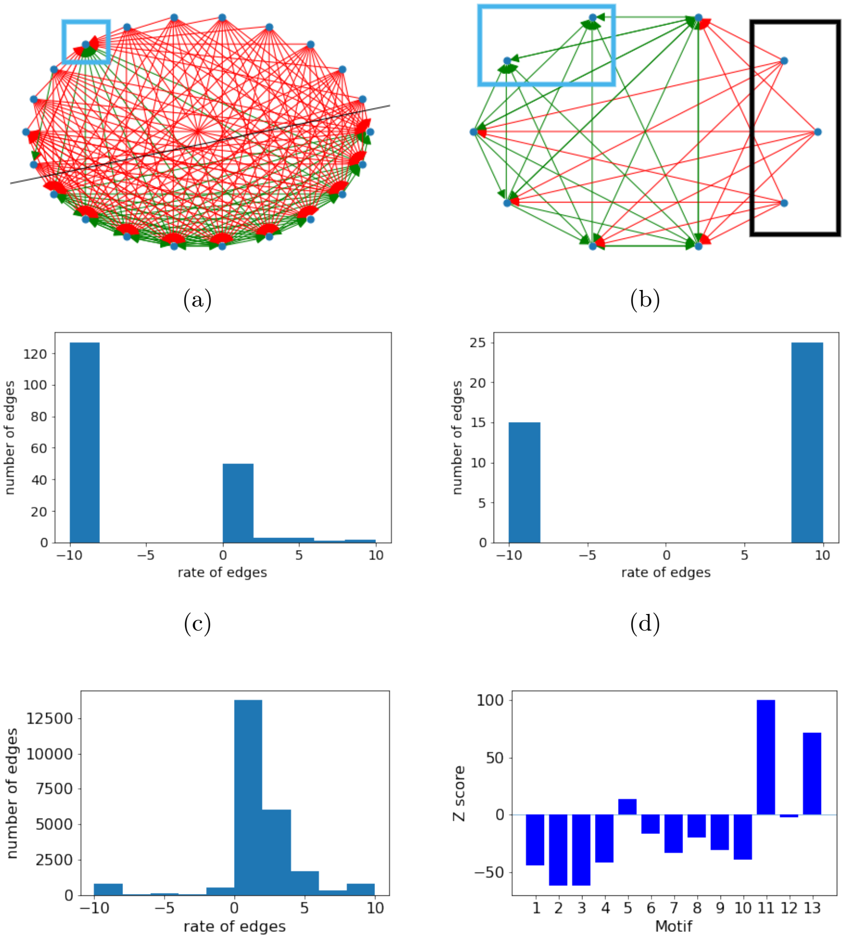
### C.1 Random graph models

Erdős, Rényi, and Gilbert introduced the notion of a random graph and set the foundations of random graph theory with two closely related models $G(n, m)$ and $G(n, p)$ [8, 6]. For simplicity, we assume that the set of nodes is $[n]$. In $G(n, m)$, a graph is sampled uniformly at random among all labeled graphs with $m$ edges, whereas in $G(n, p)$ each edge is included in the graph with probability $p$, independently from every other edge. Under mild conditions, when $p = \frac{m}{\binom{n}{2}}$ the two models are asymptotically equivalent [8]. It is common to refer to these models as Erdős- Rényi models. The $G(n, p)$ model for undirected graphs has a natural counterpart for directed graphs. Due to the simplicity of the model, fitting an Erdős- Rényi to a given directed graph $G$ with $n$ nodes and $m$ edges corresponds to sampling from $G(n, p = \frac{m}{n(n-1)})$. Exponential random graph models generalize random binomial graphs [2].

A popular random graph model as a null model in assessing the significance of motifs is the configuration model initially introduced by Bollobas [1]. Since then, numerous variants of the configuration model have been produced, e.g., [14], all under the same name. This fact has generated some confusion; see the work of Fosdick et al. [7] for a detailed discussion of the configuration model.

The Chung-Lu model is a simple, powerful variant of the configuration model where an edge exists between any two vertices with probability proportional to the product of their expected degrees [3, 4]. In our experiments we use the directed version of Chung-Lu that takes into account both the in- and out-degree sequence of the input graph $G$. Specifically, let $deg^-(v)$ and $deg^+(v)$ be the in- and out-degree of node $v$ respectively. The model creates an arc $u \to v$ with probability $deg^+(u) \cdot deg^-(v) / \sum_k deg^-(k)$; it is clear that the model preserves the degree sequences in expectation.

The edge-swap configuration model preserves the degrees in the following way: starting from a graph with the correct degree sequence, we repeatedly randomly sample two edges $(u, v), (x, y) \in E$, replace them with edges $(u, y), (x, v)$



(a) (b)

(c) (d)

(e) (f)

Fig. 1: Subgraphs found by MotifScope on (a) bitcoin-OTC, and (b) bitcoin-Alpha networks by contrasting pairs (9,11) and (4,10) respectively. Negative and positive rated edges are shown in red and green respectively. (c), (d) Histogram of edge ratings in the two found induced subgraphs respectively. (e) Histogram of edge ratings in the whole bitcoin-Alpha network. The rating histogram of Bitcoin-OTC is similar. (f) $z$-score of motifs of order 3 in the Bitcoin-alpha network.

if they are not already in the graph. It has been proved that when the degree sequence is not far from being regular, the Markov chain mixes rapidly and converges to sampling uniformly at random among all graphs with a given degree sequence [5, 11], but for skewed degree distributions that we typically observe in reality we just repeat the process as many times as possible, given our computational resources.

We also use another variation of the configuration model in our experiments for partially directed graphs [16]. The motivation behind this model is to capture better reciprocal directed edges. Specifically, the model combines every two reciprocal edges into one undirected edge in directed graph, and therefore assigns each node $v \in V$ with three degrees: the undirected degree $deg_u(v)$, the in-degree $deg^-(v)$, and the out-degree $deg^+(v)$. The model first generates undirected (i.e., bidirectional) edges based on undirected degrees using a configuration model, e.g., the Chung-Lu model. Then directed non-reciprocal edges are generated based on the directed Chung-Lu model.

We also use a variety of more recent models in our experiments. The stochastic Kronecker model [13] fits a seed matrix to the input graph $G$ as follows. We use a $2 \times 2$ seed matrix $S$,

$$S = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ where } a + b + c + d = 1, a \cdot b \cdot c \cdot d > 0.$$

Each edge is generated according to the probability matrix $P = S \otimes S \otimes \ldots \otimes S$ where $\otimes$ denotes the Kronecker product operation. Several major properties of the model are well understood [15]. Recently Koevering et al. proposed a new model, the prescribed $k$-core model [17] for undirected graphs. The model is based on a novel Markov chain that samples among all graphs with a given $k$-core value. Although the model is not as well understood theoretically as older models, it produces high-quality samples in numerous cases [17]. Unfortunately, extending the model to the directed case is not straightforward, so in our experiments we convert the directed graph into an undirected one. To obtain a directed graph, once the model generates an undirected graph, for each edge we independently sample whether it is bi-directional based on the probability of having reciprocal edges in the original graph.

Finally, deep learning based graph models have recently been proposed, as described in the comprehensive survey given by Guo and Zhao [10]. In our work we use the popular GraphRNN model [18]. The key idea of GraphRNNs is to leverage the power of recursive neural networks (RNNs) [9], by representing the graph under different node orderings as different "time series"-like sequences, and then building models to predicting occurrences of edges. We adapt this model to its directed counterpart by doubling the size of edge sequence in order to represent the existence of edges in two directions, enabling GraphRNN training and inference.

### C.2 Findings

To answer Q3, we assess the concentration using the variance divided by the empirical mean ($\frac{\sigma^2}{\mu^2}$). As we know from Chebyshev's inequality,

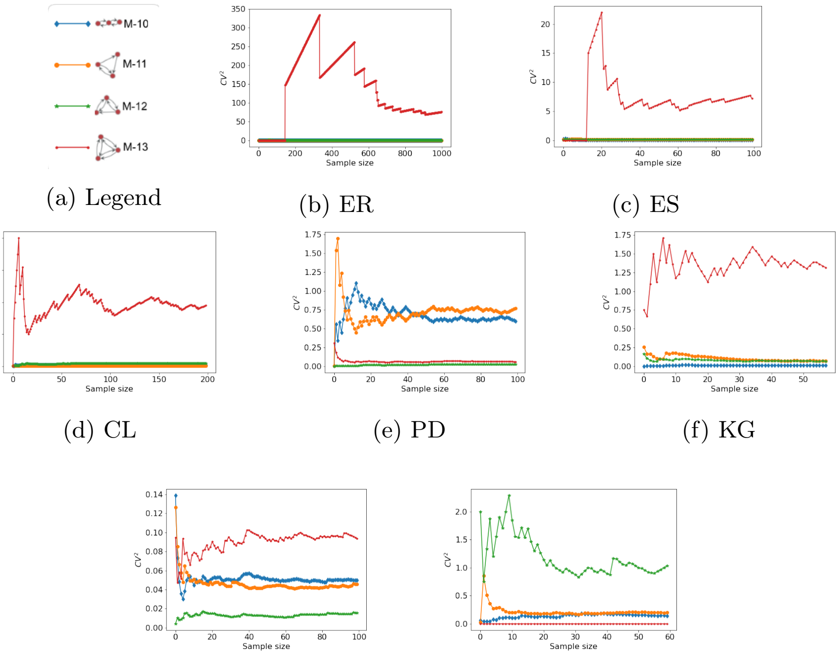
$$\mathbf{Pr}\left[|X - \mathbb{E}\left[X\right]| \geq \epsilon\mathbb{E}\left[X\right]\right] \leq \frac{Var(X)}{\epsilon^2\mathbb{E}\left[X\right]^2},$$

so a small ratio $\frac{\sigma^2}{\mu^2}$ indicates strong concentration around the true expected value. The related ratio $\frac{\sigma}{\mu}$ is known as coefficient of variation (CV). We use the notation $CV^2$ to denote the coefficient of variation squared, i.e., the ratio $\frac{\sigma^2}{\mu^2}$. Figures 2(c) through (i) show our findings for each null model. Specifically, for each model we plot $CV^2$ as a function of the sample size for up to four motifs, that are shown in the legend of Figure 2(b). As we observe, the number of required samples is both a function of the model, but also a function of the motif. This is not counter-intuitive; consider for example motif 13 for the directed Erdős-Rényi model. Since the *C. elegans* graph is sparse, the probability of three nodes forming all possible directed arcs is $p^6 \sim 1.7 \cdot 10^{-8}$. The obtained counts for motif 13 are mostly 0, with occasional 1s that are responsible for the spikes. Even when the number of samples is 1 000, we do not observe a strong concentration. Furthermore, we observe that what is a “hard” motif for most of the models (i.e., motif 13) is not hard for all; specifically the partial configuration model captures motif 13 better than motif 10.

As a rule of thumb, we observe that for most motifs a few tenths of samples are required for all models to obtain concentrated counts around the expectation. However, for certain motifs a significantly larger number of samples is required, and thus results reported in the literature that use few tens of samples should be considered with even greater caution. Finally, we observe that the recent model prescribed $k$-core due to Koevering, Benson, and Kleinberg [17] achieves the smallest values of $CV^2$ across all four motifs we consider, even with a small number of samples required, outperforming even GraphRNNs. Nonetheless, the quality of the output of the KC model (see Figure 4(g) in the main text) is questionable as it assigns negative scores to most motifs of order 3.

## References

1. Bollobás, B.: A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. European Journal of Combinatorics **1**(4), 311–316 (1980)
2. Chatterjee, S., Diaconis, P.: Estimating and understanding exponential random graph models. The Annals of Statistics **41**(5), 2428–2461 (2013)
3. Chung, F., Chung, F.R., Graham, F.C., Lu, L., Chung, K.F., et al.: Complex graphs and networks. No. 107, American Mathematical Soc. (2006)
4. Chung, F., Lu, L.: The average distances in random graphs with given expected degrees. PNAS **99**(25), 15879–15882 (2002). https://doi.org/10.1073/pnas.252631999
5. Cooper, C., Dyer, M., Handley, A.J.: The flip markov chain and a randomising p2p protocol. In: Proceedings of the 28th ACM PODC. pp. 141–150 (2009)



(a) Legend

(b) ER

(c) ES

(d) CL

(e) PD

(f) KG

(g) KC

(h) GR

Fig. 2: (a) Motifs' legend. (b)-(h) Coefficient of variation squared ($\mathrm{CV}^2$) for various motifs vs. the number of sampled graphs from each null model.

6. Erdős, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci **5**(1), 17–60 (1960)
7. Fosdick, B.K., Larremore, D.B., Nishimura, J., Ugander, J.: Configuring random graph models with fixed degree sequences. Siam Review **60**(2), 315–355 (2018)
8. Frieze, A., Karoński, M.: Introduction to random graphs. Cambridge University Press (2016)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
10. Guo, X., Zhao, L.: A systematic survey on deep generative models for graph generation (2020)
11. Kannan, R., Tetali, P., Vempala, S.: Simple markov-chain algorithms for generating bipartite graphs and tournaments. Random Struct. Algorithms **14**(4), 293–308 (1999)
12. Kumar, S., Spezzano, F., Subrahmanian, V., Faloutsos, C.: Edge weight prediction in weighted signed networks. In: ICDM. pp. 221–230. IEEE (2016)
13. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker graphs: An approach to modeling networks. J. Mach. Learn. Res (JMLR) **11**, 985–1042 (2010)
14. Newman, M.: Networks: An Introduction. OUP Oxford (2010), `https://books.google.com/books?id=LrFaU4XCsUoC`
15. Seshadhri, C., Pinar, A., Kolda, T.G.: An in-depth analysis of stochastic kronecker graphs. Journal of the ACM (JACM) **60**(2), 1–32 (2013)
16. Spricer, K., Britton, T.: The configuration model for partially directed graphs. Journal of Statistical Physics **161**, 965–985 (2015)
17. Van Koevering, K., Benson, A., Kleinberg, J.: Random graphs with prescribed k-core sequences: A new null model for network analysis. In: Proceedings of the Web Conference 2021. p. 367–378. WWW '21 (2021)
18. You, J., Ying, R., Ren, X., Hamilton, W.L., Leskovec, J.: Graphrnn: Generating realistic graphs with deep auto-regressive models. In: ICML (2018)