On the Power of Edge Independent Graph Models

Sudhanshu Chanpuriya  
University of Massachusetts Amherst  
schanpuriya@umass.edu

Cameron Musco  
University of Massachusetts Amherst  
cmusco@cs.umass.edu

Konstantinos Sotiropoulos  
Boston University  
ksotiro@bu.edu

Charalampos E. Tsourakakis  
Boston University & ISI Foundation  
tsourolampis@gmail.com

Abstract

Why do many modern neural-network-based graph generative models fail to reproduce typical real-world network characteristics, such as high triangle density? In this work we study the limitations of edge independent random graph models, in which each edge is added to the graph independently with some probability. Such models include both the classic Erdős-Rényi and stochastic block models, as well as modern generative models such as NetGAN, variational graph autoencoders, and CELL. We prove that subject to a bounded overlap condition, which ensures that the model does not simply memorize a single graph, edge independent models are inherently limited in their ability to generate graphs with high triangle and other subgraph densities. Notably, such high densities are known to appear in real-world social networks and other graphs. We complement our negative results with a simple generative model that balances overlap and accuracy, performing comparably to more complex models in reconstructing many graph statistics.

1 Introduction

Our work centers on edge independent graph models, in which each edge \((i, j)\) is added to the graph independently with some probability \(P_{ij} \in [0, 1]\). Formally,

**Definition 1 (Edge Independent Graph Model).** For any symmetric matrix \(P \in [0, 1]^{n \times n}\) let \(G(P)\) be the distribution over undirected unweighted graphs where \(G \sim G(P)\) contains edge \((i, j)\) independently, with probability \(P_{ij}\). I.e., \(p(G) = \prod_{(i, j) \in E(G)} P_{ij} \cdot \prod_{(i, j) \notin E(G)} (1 - P_{ij})\).

Edge independent models encompass many classic random graph models. This includes the Erdős-Rényi model, where for all \(i \neq j\), \(P_{ij} = p\) for some fixed \(p \in [0, 1]\) [10]. It also includes the stochastic block model where \(P_{ij} = p\) if two nodes are in the same community and \(P_{ij} = q\) if two nodes are in different communities for some fixed \(p, q \in [0, 1]\) with \(q < p\) [29]. Other examples include e.g., the Chung-Lu configuration model [5], stochastic Kronecker graphs [17].

Recently, significant attention has focused on graph generative models, which seek to learn a distribution over graphs that share similar properties to a given training graph, or set of graphs. Many algorithms parameterize this distribution as an edge independent model or closely related distribution. E.g., NetGAN and the closely related CELL model both produce \(P \in [0, 1]^{n \times n}\) and then sample edges independently without replacement with probabilities proportional to its entries, ensuring that at least one edge is sampled adjacent to each node [3] [23]. Variational Graph Autoencoders (VGAE), GraphVAE, Graphite, and MolGAN are also all based on edge independent models [16] [28] [8] [13].

Given their popularity in both classical and modern graph generative models, it is natural to ask:
How suited are edge independent models to modeling real-world networks. Are they able to capture features such as power-law degree distributions, small-world properties, and high clustering coefficients (triangle densities)?

1.1 Impossibility Results for Edge Independent Models

In this work we focus on the ability of edge independent models to generate graphs with high triangle, or other small subgraph densities. High triangle density (equivalently, a high clustering coefficient) is a well-known hallmark of real-work networks \cite{31, 25, 9} and has been the focus of recent work exploring the power and limitations of edge-independent graph models \cite{27, 4}.

It is clear that edge independent models can generate triangle dense graphs. In particular, \( P \in [0,1]^{n \times n} \) in Def. 1 can be set to the binary adjacency matrix of any undirected graph, and \( G(P) \) will generate that graph with probability 1, no matter how triangle dense it is. However, this would not be a particularly interesting generative model – ideally \( G(P) \) should generate a wide range of graphs. To capture this intuitive notion, we define the overlap of an edge-independent model, which is closely related to the overlap stopping criterion for training used in training graph generative models \cite{3, 23}.

**Definition 2 (Expected Overlap).** For symmetric \( P \in [0,1]^{n \times n} \) let \( V(P) \) \( \text{def} \frac{E_{G_1 \sim G(P)} |E(G_1) \cap E(G_2)|}{V(P)} \).

That is, for any \( P \in [0,1]^{n \times n} \), \( Ov(P) \in [0,1] \) is the ratio of the expected number of edges shared by two graphs drawn independently from \( G(P) \) to the expected number of edges in a graph drawn from \( G(P) \). In one extreme, when \( P \) is a binary adjacency matrix, \( Ov(P) = 1 \), and our generative model has simply memorized a single graph. In the other, if \( P_{ij} = p \) for all \( i \neq j \) (i.e., \( G(P) \) is Erdös-Rényi), \( Ov(P) = p \). This is the minimum possible overlap when \( V(P) = p \cdot \binom{n}{2} \).

Our main result is that for any edge independent model with bounded overlap, \( G \sim G(P) \) cannot have too many triangles in expectation. In particular:

**Theorem 1 (Main Result – Expected Triangles).** For a graph \( G \), let \( \Delta(G) \) denote the number of triangles in \( G \). Consider symmetric \( P \in [0,1]^{n \times n} \).

\[
E_{G \sim G(P)}[\Delta(G)] \leq \frac{\sqrt{2}}{3} \cdot Ov(P)^{3/2} \cdot V(P)^{3/2}.
\]

As an example, consider the setting where the distribution generates sparse graphs, with \( V(P) = \Theta(n) \). Theorem 1 shows that whenever \( Ov(P) = o(1/n^{1/3}) \), \( E_{G \sim G(P)}[\Delta(G)] = o(n) \) – i.e. the graph is very triangle sparse with the number of triangles sublinear in the number of nodes. This verifies that an Erdös-Rényi graph cannot achieve simultaneously linear number of edges (i.e., \( Ov(P) = O(1/n) \)) and super-linear number of triangles (i.e., \( Ov(P) = \Omega(1/n^{1/3}) \)) under our proposed lens of viewing generative models.

We extend Theorem 1 to give similar bounds for the density of squares and other \( k \)-cycles (Thm. 4), as well as for the global clustering coefficient (Thm. 6). In all cases we show that our bounds are tight – e.g., in the triangle case, there is indeed an edge independent model with \( E_{G \sim G(P)}[\Delta(G)] = \Theta \left( Ov(P)^{3/2} \cdot V(P)^{3/2} \right) \), matching the lower bound in Theorem 1.

1.2 Empirical Findings

Our theoretical results help explain why, despite performing well in a variety of other metrics, edge independent graph generative models have been reported to generate graphs with many fewer triangles and squares on average than the real-world graphs that they are trained on. Rendsburg et al. \cite{23} test a suite of these models, including their own CELL model and the related NetGAN model \cite{4}. Of all these models, when trained on the CORA-ML graph with 2,802 triangles and 14,268 squares, none is able to generate graphs with more than 1,461 triangles and 6,880 squares on average. Similar gaps are observed for a number of other graphs. Rendsburg et al. also report that the triangle count increases as their notion of overlap (closely related to Def. 2) increases. Theorem 1 demonstrates that this underestimation of triangle count, and its connection to overlap is inherent to all edge independent models, no matter how refined a method used to learn the underlying probability matrix \( P \).
While our theoretical results bound the performance of any edge independent model, there may still be variation in how specific models trade-off overlap and realistic graph generation. To better understand this trade-off, we introduce two simple models with easily tunable overlap as baselines. One is based on reproducing the degree sequence of the original graph; the other, which is even simpler, is based on reproducing the volume. In both models, $P$ is a weighted average of the input graph adjacency matrix and a probability matrix of minimal complexity which matches either the input degrees or the volume. In the latter case, to match just the volume, we simply use an Erdős-Rényi graph. In the former case, to match the degree sequence, we introduce our own model, the odds product model; this model is similar to the Chung-Lu configuration model [5], but, unlike Chung-Lu, is able to match degree sequences of real-world graphs with high maximum degree. We find that these simple baselines are often competitive with more complex models like CELL in terms of matching key graph statistics, like triangle count and clustering coefficient, at similar levels of overlap.

1.3 Related Work

Existing impossibility results. Our work is inspired by that of Seshadhri et al. [27], which also proves limitations on the ability of edge independent models to represent triangle dense graphs. They show that if $P = \max(0, \min(1, XX^T))$ where $X \in \mathbb{R}^{n \times k}$ for $k \ll n$ and the max and min are applied entrywise, then $G \sim G(P)$ cannot have many triangles adjacent to low-degree nodes in expectation. This setting arises commonly when $P$ is generated using low-dimensional node embeddings – represented by the rows of $X$. Chanpuriya et al. [4], show that in a slightly more general model, where $P = \max(0, \min(1, XY^T))$, this lower bound no longer holds – $X, Y \in \mathbb{R}^{n \times k}$ can be chosen so that $P$ is the binary adjacency matrix of any graph with maximum degree upper bounded by $O(k)$ – no matter how triangle dense that graph is. Thus, even such low-rank edge independent models can represent triangle dense graphs – by memorizing a single one. In the appendix, we prove a similar result when $P$ is generated from the CELL model of [23], which simplifies NetGAN [3].

Our results show that this trade-off between the ability to capture triangle density and memorization is inherent – even without any low-rank constraint, edge independent models with low overlap simply cannot represent graphs with high triangle or other small subgraph density.

It is well understood that specific edge independent models, e.g., Erdős-Rényi graphs, the Chung-Lu model, and stochastic Kronecker graphs, do not capture many properties of real-world networks, including high triangle density [31], [22]. Our results can be viewed as a generalization of these observations, to all edge independent models with low overlap. Despite the limitations of classic models, edge independent models are still very prevalent in today’s literature on graph generative models. Our more general results make clear the limitations of this approach.

Non-independent models. While edge independent models are very prevalent in the literature, many important models do not fit into this framework. Classic models include the Barabási–Albert and other preferential attachment models [2], Watts–Strogatz small-world graphs [31], and random geometric graphs [6]. Many of these models were introduced directly in response to shortcomings of classic edge independent models, including their inability to produce high triangle densities.

More recent graph generative models include GraphRNN [12] and a number of other works [19], [20]. Our impossibility results do not apply to such models, and in fact suggest that perhaps they may be preferable to edge independent models, if a distribution over graphs with high triangle density is desired. A very interesting direction for future work would be to prove limitations on broad classes of non-independent models, and perhaps to understand exactly what type of correlation amongst edges is needed to generate graphs with both low overlap and hallmark features of real-world networks.

2 Impossibility Results for Edge Independent Models

We now prove our main results on the limitations of edge independent models with bounded overlap. We start with a simple lemma that will be central in all our proofs.

Lemma 2. For any symmetric $P \in [0, 1]^{n \times n}$, \[ \frac{\|P\|^2_F}{2} \leq Ov(P) \cdot V(P) \leq \|P\|^2_F. \]

1We note that for non-edge independent models, the measure of overlap as defined earlier should be adapted to take into account the order (permutation) of the vertices in the final graph. In particular, the overlap in this case should be the maximum value of it over any permutation of the vertex set.
We note that another example when Theorem 1 is tight is when $P$ is simply an Erdős–Rényi graph.

**Theorem 3 (Tightness of Expected Triangle Bound).** For any $\gamma \in (0, 1]$, there exists a symmetric $P \in [0, 1]^{n \times n}$ with $Ov(P) = \gamma$ and $\mathbb{E}_{G \sim \mathcal{G}(P)}[\Delta(G)] = \Theta(\gamma^{3/2} \cdot V(P)^{3/2})$.

**Proof.** Let $P_{ij} = \gamma$ for all $i \neq j$. We have $V(P) = \gamma \cdot \binom{n}{3}$ and $Ov(P) \cdot V(P) = \gamma^2 \cdot \binom{n}{2}$ Thus, $Ov(P) = \gamma$. Further, by linearity of expectation,

$$
\mathbb{E}_{G \sim \mathcal{G}(P)}[\Delta(G)] = \gamma^3 \cdot \binom{n}{3} = \Theta(\gamma^3 \cdot n^3) = \Theta(\gamma^{3/2} \cdot V(P)^{3/2}).
$$

The bound of Theorem 3 is tight when $P$ is a union of a fixed clique on $\Theta(\gamma \cdot n)$ nodes and an Erdős–Rényi graph with connection probability $1/n$ on the rest of the nodes.
2.2 Squares and Other $k$-cycles

We can extend Thm. 1 to bound the expected number of $k$-cycles in $G \sim \mathcal{G}(P)$ in terms of $Ov(P)$.

**Theorem 4 (Bound on Expected $k$-cycles).** For a graph $G$, let $C_k(G)$ denote the number of $k$-cycles in $G$. Consider symmetric $P \in [0,1]^{n \times n}$.

\[
E_{G \sim \mathcal{G}(P)} [C_k(G)] \leq \frac{2k/2}{2k} \cdot Ov(P)^{k/2} \cdot V(P)^{k/2}.
\]

**Proof.** For notational simplicity, we focus on $k = 4$. The proof directly extends to general $k$. $C_4(G)$ is the number of non-backtracking 4-cycles in $G$ (i.e. squares), which can be written as

\[
E_{G \sim \mathcal{G}(P)} [C_4(G)] = \frac{1}{8} \cdot \sum_{i,j \in [n]} \sum_{1 \leq i,j \leq n} \sum_{\ell \in [n]} \sum_{1 \leq i,j,k} P_{ij}P_{jk}P_{kl}P_{li}.
\]

The $1/8$ factor accounts for the fact that in the sum, each square is counted 8 times – once for each potential starting vector $i$ and once for each direction it may be traversed. For general $k$-cycles this factor would be $\frac{1}{2k}$. We then can bound

\[
E_{G \sim \mathcal{G}(P)} [C_4(G)] \leq \frac{1}{8} \cdot \sum_{i,j \in [n]} \sum_{1 \leq i,j \leq n} \sum_{\ell \in [n]} \sum_{1 \leq i,j,k} P_{ij}P_{jk}P_{kl}P_{li} = \frac{1}{8} \cdot \text{tr}(P^4).
\]

For general $k$-cycles this bound would be $E_{G \sim \mathcal{G}(P)} [C_k(G)] \leq \frac{1}{2k} \cdot \text{tr}(P^k)$. This in turn gives

\[
E_{G \sim \mathcal{G}(P)} [C_k(G)] \leq \frac{1}{2k} \cdot |\lambda_1(P)|^{k-2} \cdot \|P\|^2_F \leq \frac{1}{2k} \cdot \|P\|^k_F \leq \frac{2k/2}{2k} \cdot Ov(P)^{k/2} \cdot V(P)^{k/2},
\]

where the last bound follows from Lemma 2. This completes the theorem. \(\square\)

It is not hard to see that Theorem 4 is also tight up to a constant depending on $k$ for any overlap $\gamma \in (0, 1]$, also for an Erdős-Rényi graph with connection probability $\gamma$.

**Theorem 5 (Tightness of Expected $k$-cycle Bound).** For any $\gamma \in (0, 1]$, there exists $P \in [0,1]^{n \times n}$ with $Ov(P) = \gamma$ and $E_{G \sim \mathcal{G}(P)} [C_k(G)] = \Theta \left( \frac{k^{k/2} \cdot V(P)^{k/2}}{k!} \right)$.

2.3 Clustering Coefficient

Theorem 1 shows that the expected number of triangles generated by an edge independent model is bounded in terms of the model’s overlap. Intuitively, we thus expect that graphs generated by the edge independent model will have low global clustering coefficient, which is the fraction of wedges in the graph that are closed into triangles.

**Definition 3 (Global Clustering Coefficient).** For a graph $G$ with $\Delta(G)$ triangles, no self-loops, and node degrees $d_1, d_2, \ldots, d_n$, the global clustering coefficient is given by

\[
C(G) = \frac{3 \Delta(G)}{\sum_{i=1}^{n} d_i(d_i-1)}.
\]

We extend Theorem 1 to give a bound on $E_{G \sim \mathcal{G}(P)} [C(G)]$ in terms of $Ov(P)$. The proof, given in Appendix A, is related, but more complex due to the $\sum_{i=1}^{n} d_i(d_i-1)$ in the denominator of $C(G)$.

**Theorem 6 (Bound on Expected Clustering Coefficient).** Consider symmetric $P \in [0,1]^{n \times n}$ with zeros on the diagonal and with $V(P) \geq 2n$.

\[
E_{G \sim \mathcal{G}(P)} [C(G)] = O \left( \frac{Ov(P)^{3/2} \cdot n}{V(P)^{1/2}} \right).
\]

Thus, to have a constant clustering coefficient for a graph with $O(n)$ edges in expectation, we need $Ov(P) = \Omega(1/n^{1/3})$. Note that the requirement of $V(P) \geq 2n$ is very mild – it means that the expected average degree is at least 1.

As with our triangle bound, Theorem 6 is tight when $\mathcal{G}(P)$ is just an Erdős-Rényi distribution.
**Theorem 7** (Tightness of Expected Clustering Coefficient Bound). For any \( \gamma \in (0,1] \), there exists \( P \in [0,1]^{n \times n} \) with zeros on the diagonal, \( \text{Ov}(P) \leq \gamma \) and \( \mathbb{E}_{G \sim \mathcal{G}(P)}[C(G)] = \Theta \left( \frac{\gamma^{3/2} \cdot n}{V(P)^{1/2}} \right) \).

**Proof.** Let \( P_{ij} = \gamma \) for all \( i \neq j \). We have \( V(P) = \gamma \cdot \binom{n}{2} = \Theta(\gamma n^2) \) and \( \text{Ov}(P) = \gamma \). Additionally, \( \mathbb{E}[\Delta(G)] = \Theta(\gamma^3 \cdot n^3) \), and, if \( n \) is large enough with respect to \( \gamma \), with very high probability, \( \sum_{i=1}^{n} d_i(d_i-1) \leq \sum_{i=1}^{n} d_i^2 = O(\gamma^2 n^3) \). This gives:

\[
\mathbb{E}_{G \sim \mathcal{G}(P)}[C(G)] = \Theta(\gamma) = \Theta \left( \frac{\gamma^{3/2} \cdot n}{\gamma^{1/2} \cdot n} \right) = \Theta \left( \frac{\gamma^{3/2} \cdot n}{V(P)^{1/2}} \right).
\]

\( \square \)

### 3 Baseline Edge Independent Models

We now shift from proving theoretical limitations of edge independent models to empirically evaluating the tradeoff between overlap and performance for a number of particular models. Given an input adjacency matrix \( A \in \{0,1\}^{n \times n} \), these generative models produce a \( P \in [0,1]^{n \times n} \), samples from which should match various graph statistics of \( A \), such as the triangle count, clustering coefficient, and assortativity. At the same time, \( P \) should ideally have lower overlap so that the model does not just memorize the original graph. We propose two simple generative models as baselines to more complicated existing models – in both the level of overlap is easily tuned. Our first baseline, the **odds product model**, is based on just matching the degree sequence of \( A \); more simple still, the second baseline computes \( P \) as a linear function of \( A \), just memorizing the original graph. We do not have a proof that Algorithm 1 always converges and proving that it always converges would be an interesting future direction.

**Odds product model.** In this model, each node is assigned a logit \( \ell \in \mathbb{R} \), and the probability of adding an edge between nodes \( i \) and \( j \) is \( P_{ij} = \sigma(\ell_i + \ell_j) \), where \( \sigma \) is the logistic function. We fit the model by finding a vector \( \ell \in \mathbb{R}^n \) of logits, with one logit for each node, such that the reconstructed network has the same expected degrees (i.e. row and column sums) as the original graph. We note that this model can be seen as a special case of the MaxEnt \([7]\) and random-effects \([21,14,15]\) models. In the context of directed graphs, \( \ell_i \) has been called the expansiveness or popularity of node \( i \) \([12]\).

For adjacency matrix \( A \in \{0,1\}^{n \times n} \), we denote its degree sequence by \( d = A1 \in \mathbb{R}^n \), where \( 1 \) is the all-ones vector of length \( n \). Similarly, the degree sequence of the model is \( \hat{d} = P1 \). We pose fitting the model as a root-finding problem: we seek \( \ell \in \mathbb{R}^n \) such that the degree errors are zero, that is, \( \hat{d} - d = 0 \). We use the multivariate Newton-Raphson method to solve this root-finding problem. To apply Newton-Raphson, we need the Jacobian matrix \( J \) of derivatives of the degree errors with respect to the entries of \( \ell \). Since \( \hat{d} \) does not vary with \( \ell \), these derivatives are exactly \( \frac{\partial \hat{d}}{\partial \ell} \). Letting \( \delta_{ij} \) be 1 if \( i = j \) and 0 otherwise (i.e. the Kronecker delta), we compute in Appendix A:

\[
\frac{\partial \hat{d}}{\partial \ell_j} = P_{ij} (1 - P_{ij}) + \delta_{ij} \sum_{k \in [n]} P_{ik} (1 - P_{ik}) \cdot
\]

In Algorithm 1 we provide pseudocode for the fully Jacobian matrix \( J \) and for implementing Newton-Raphson method to compute \( P \). We do not have a proof that Algorithm 1 always converges and produces \( \ell \) which exactly reproduces in the inut degree sequence. However, the algorithm converged on all test cases, and proving that it always converges would be an interesting future direction.

Our odds product model can be viewed as a variant of the Chung-Lu configuration model \([5]\), which is also based on degree sequence matching. However, our model comes without a certain restriction on the maximum degree: in Chung-Lu, it is assumed that the degrees of all nodes are bounded above by the square root of the volume of the graph, that is, \( d_i \leq \sqrt{V(A)} \) for all nodes \( i \). Given this restriction, each node is assigned a weight \( w_i = d_i/\sqrt{V(A)} \), and the probability of adding edge \((i,j)\) is \( P_{ij} = w_i w_j \). Since the weights are all in \([0,1]\), they can be interpreted as probabilities, and the probability of adding an edge between two nodes is the product of the two nodes’ probabilities.

Our odds product model works similarly, but instead of a probability, for each node, there is an associated odds, that is, a value in \((0,\infty)\), and the odds of adding an edge between two nodes is the product of the two nodes’ odds. There is a one-to-one-to-one relationship between probability \( p \in [0,1] \), odds \( o = 1/p \in [0,\infty) \), and logit \( \ell = \ln(o) \in (-\infty, +\infty) \). We outlined above how our
We compare our proposed models from Section 3 with a number of existing models described below.

4 Experimental Results

We now present our evaluations of different edge independent graph generative models in terms of the tradeoff achieved between overlap and performance in generating graphs with similar key statistics to an input network. These experiments highlight the strengths and limitations of each model, as well as the overall limitations of this class, as established by our theoretical bounds.

4.1 Methods

We compare our proposed models from Section 3 with a number of existing models described below.

1. CELL [23] (Cross-Entropy Low-rank Logits) An alternative to the popular NetGAN method [3] which strips the proposed architecture of deep leaning components and achieves comparable performance in significantly less time, via a low-rank approximation approach. To control overlap, we follow the approach of the original paper, halting training once the generated graph exceeds a specified overlap threshold with the input graph. We set the rank parameter to a value that allows us to get up to 75% overlap (typical values are 16 and 32).

2. TSVD (Truncated Singular Value Decomposition) A classic spectral method which computes a rank-$k$ approximation of the adjacency matrix using truncated SVD. As in [27], the resulting matrix is clipped to $[0,1]$ to yield $P$. Overlap is controlled by varying $k$.

model is based on adding logits associated with each node; since the odds is the exponentiation of the logit, the model can equally be viewed as multiplying odds associated with nodes.

Varying overlap in the odds product model. We propose a simple method to control the trade-off between overlap and accuracy in matching the input graph statistics in the odds product model. Given the original adjacency matrix $A$ and the $P$ generated by the odds product model to match the degree sequence of $A$, we use a convex combination of $P$ and $A$. That is, we use $P = (1 - \omega)P + \omega A$, where $0 \leq \omega \leq 1$. As $\omega$ increases to 1, $P$ approaches a model which returns the original graph with high certainty; hence high $\omega$ produce $P$ with high overlap which closely match graph statistics, while low $\omega$ produce $P$ with lower overlap which may diverge from $A$ in some statistics. Note that since $P$ is a convex combination of adjacency matrices with the expected degree sequence of $A$, $P$ also has the same expected degree sequence regardless of the value of $\omega$.

Linear model. As an even simpler baseline, we also propose and evaluate the following model: we produce an Erdös-Rényi model $P$ with the same expected volume as the original graph $A$. Then return a rank-$k$ approximation $\tilde{P}$ of $P$ and $A$. In particular, each entry of $P$ is $V(A)/n^2$, and, as with the odds product model, $\tilde{P} = (1 - \omega)P + \omega A$, where $0 \leq \omega \leq 1$. This model can alternatively be seen as producing a $P$ by lowering each entry of $A$ which is 1 to some probability $\alpha$, and raising each entry of $A$ which is 0 to a probability $\beta$, with $\alpha \geq \beta$, such that the volume is conserved.

Algorithm 1 Fitting the odds product model

```
Algorithm 1 Fitting the odds product model
input graphical degree sequence $d \in \mathbb{R}^n$, error threshold $\epsilon$
output symmetric matrix $P \in (0,1)^{n \times n}$ with row/column sums approximately $d$
1: $\ell \leftarrow 0$
2: $P \leftarrow \sigma (\ell^T + 1^T \ell^T)$ \hspace{1cm} $\triangleright \ell \in \mathbb{R}^n$ is the vector of logits, initialized to all zeros
3: $\tilde{d} \leftarrow P 1$ \hspace{1cm} $\triangleright$ degree sequence of $P$
4: while $\|\tilde{d} - d\|_2 > \epsilon$ do
5: $B \leftarrow P \odot (1^T - P)$ \hspace{1cm} $\triangleright \odot$ is an entrywise product
6: $J \leftarrow B + \text{diag}(B1)$ \hspace{1cm} $\triangleright$ diag is the diagonal matrix with the input vector along its diagonal
7: $\ell \leftarrow \ell - J^{-1}(\tilde{d} - d)$ \hspace{1cm} $\triangleright$ rather than inverting $J$, we solve this linear system
8: $P \leftarrow \sigma (\ell^T + 1^T \ell^T)$
9: $\tilde{d} \leftarrow P 1$
10: end while
11: return $P$
```
Table 1: Dataset summaries

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th>Edges</th>
<th>Triangles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CITESEER [26]</td>
<td>2,110</td>
<td>7,336</td>
<td>1,083</td>
</tr>
<tr>
<td>CORA [26]</td>
<td>2,485</td>
<td>10,138</td>
<td>1,558</td>
</tr>
<tr>
<td>ROAD-MINNESOTA [24]</td>
<td>2,640</td>
<td>6,604</td>
<td>53</td>
</tr>
<tr>
<td>PPI [30]</td>
<td>3,852</td>
<td>75,682</td>
<td>91,461</td>
</tr>
<tr>
<td>FACEBOOK [18]</td>
<td>4,039</td>
<td>176,468</td>
<td>1,612,010</td>
</tr>
</tbody>
</table>

3. **CCOP** (Convex Combination Odds Product) The odds product model as of Sec. 3 with overlap controlled by taking a convex combination of $P$ and the input adjacency matrix $A$.

4. **HDOP** (Highest Degree Odds Product) The odds product model, but with overlap controlled by fixing the edges adjacency to a certain number of the highest degree nodes. See Appendix for results on other variants, e.g., where some number of dense subgraphs are fixed.

5. **Linear** The convex combination between the input adjacency matrix and an Erdős-Rényi graph, as described in Sec. 3 with overlap controlled by varying the $\omega$ parameter.

CCOP, HDOP, and Linear all produce edge probability matrices $P$ with the same volume, $V(G)$, in expectation as the original adjacency matrix. For TSVD, letting $L$ be the low-rank approximation of the adjacency matrix, we learn a scalar shift parameter $\sigma$ using Newton’s method such that $P = \max(0, \min(1, L + \sigma))$ has volume $V(G)$. We then generate new networks from the edge independent distribution $G(P)$ (Def. 1). For CELL, we follow the authors’ approach of generating $V(G)$ edges without replacement - an edge $(i,j)$ is added with probability proportional to $P_{ij}$.

We sample 5 networks from each distribution and report the average for every statistic. For implementation details and code, see the supplemental material.

### 4.2 Datasets and network statistics

For evaluation we use seven popular datasets with varied structure, from triangle-rich social networks to planar road networks – see Table 1. We treat each network as undirected and keep its largest connected component. In the main text we present results on three of the networks, POLBLOGS, CITESEER, and WEB-EDU. POLBLOGS is a collection of political blogs and the links between them. CITESEER is a graph of papers from six scientific categories and the citations among them. Finally, WEB-EDU is a web-graph from educational institutions. Descriptions of other networks and results on them are deferred to the appendix.

We evaluate performance in matching the following key network statistics:

1. Pearson correlation of the degree sequences of the input and the generated network.
2. Maximum degree over all nodes.
3. Exponent of a power-law distribution fit to the degree sequence.
4. Assortativity, a measure that captures the preference of nodes to attach to others with similar degree (ranging from -1 to 1).
5. Pearson correlation of the triangle sequence (number of triangles a node participates in).
6. Total triangle count (analyzed theoretically in Thm. 1).
7. Global clustering coefficient (defined in Def. 3 and analyzed theoretically in Thm. 6).
8. Characteristic path length (average path length between any two nodes).

### 4.3 Results

The theoretical results from Section 2 highlight a key weakness of edge independent generative models: they cannot generate many triangles (or other higher-order locally dense areas), without
having high overlap and thus not generating a diversity of graphs. We observe that these theoretical
findings hold in practice—generally speaking, all models tested tend to significantly underestimate
triangle count and global clustering coefficient, as well as inaccurately match the triangle degree
sequence, when overlap is low. See Figures 1, 2, and 3 for results on the POLBLOGS, CITESEER,
and WEB-EDU networks. As overlap increases, performance in reconstructing these metrics does as well,
as expected.

All methods are able to capture certain network characteristics accurately, even at low overlap. Even
for a relatively small overlap (less than 0.2), the CCOP and HDOP methods accurately capture the
degree sequences of the true networks (as they are designed to do). These methods, especially HDOP
which fixes edges from high degree nodes, often outperform more sophisticated methods like CELL
in terms of triangle density and triangle degree sequence correlation. On the other hand, CELL seems
to do a somewhat better job capturing global features, like the characteristic path length. TSVD
provides a fair compromise—it performs better than CELL in terms of degree sequence and triangle
counts, but worse in terms of characteristic path length. In general, it is the method that gives the best
results when the overlap is extremely small, appearing to be less sensitive to the variation in overlap.

Broadly speaking, all methods do reasonably well in matching the power-law degree distribution of
the networks, even when they do not match the actual degree sequence closely. With the exception
of WEB-EDU, they tend to underestimate the characteristic path length (see additional plots in the
supplemental). This is perhaps not surprising due to the independent random edge connections,
hence it would be interesting to understand more theoretically.

Figure 1: Metrics for POLBLOGS.

5 Conclusion

Our theoretical results prove limitations on the ability of any edge independent graph generative
model to produce networks that match the high triangle densities of real-world graphs, while still
generating a diverse set of networks, with low model overlap. These results match empirical findings
that popular edge independent models indeed systematically underestimate triangle density, clustering
coefficient, and related measures. Despite the popularity of edge independent models, many non-
independent models, such as graph RNNs [32] have been proposed. An interesting future direction
would be to study the representative power and limitations of such models, giving general theoretical
results that provide a foundation for the study of graph generative models.
Acknowledgments and Disclosure of Funding

CT acknowledges support from Intesa Sanpaolo Innovation Center. CM was partially supported by NSF grants no. 1763618 and 2046235 and an Adobe Research Grant. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.
References


A \hspace{1em} \text{Omitted Proofs}

**Theorem 6** (Bound on Expected Clustering Coefficient). Consider symmetric $P \in [0, 1]^{n \times n}$ with zeros on the diagonal and with $V(P) \geq 2n$.

\[ E_{G \sim G(P)}[C(G)] = O \left( \frac{Ov(P)^{3/2} \cdot n}{V(P)^{1/2}} \right). \]

**Proof.** By Theorem 1 we have $E_{G \sim G(P)}[3 \Delta(G)] \leq \sqrt{2} \cdot Ov(P)^{3/2} \cdot V(P)^{3/2}$. We will show that with high probability, $\sum_{i=1}^{n} d_i(d_i - 1) = \Omega(V(P)^2/n)$, which will give the theorem. Note that $E_{G \sim G(P)}[\sum_{i=1}^{n} d_i] = E_{G \sim G(P)}[2\cdot E(G)] = 2V(P)$. Thus, by a Bernstein bound, for large enough $n$ since $V(P) \geq 2n$,

\[ \Pr \left[ \left| \sum_{i=1}^{n} d_i - 2V(P) \right| \geq V(P)/5 \right] \leq 2 \exp \left( -\frac{V(P)^2/50}{V(P) + V(P)/15} \right) \ll \frac{1}{n^2}, \]

We can bound $\sum_{i=1}^{n} d_i^2 \geq \frac{(\sum_{i=1}^{n} d_i)^2}{n}$. Thus, with probability $\geq 1 - 1/n^2$,

\[ \sum_{i=1}^{n} d_i(d_i - 1) \geq \frac{(8/5)^2 \cdot V(P)^2}{n} - \frac{12}{5} V(P) \geq \frac{V(P)^2}{n}, \]

where in the last step we use that $V(P) \geq 2n$ and so $\frac{12}{5} \cdot V(P) \leq \frac{6}{5} \cdot V(P)^2/n$. Combined with our bound on $E_{G \sim G(P)}[3 \Delta(G)]$, and the fact that $C(G) \leq 1$ always, we have

\[ E_{G \sim G(P)}[C(G)] = O \left( \frac{Ov(P)^{3/2} V(P)^{3/2}}{V(P)/n^2} + \frac{1}{n^2} \right) = O \left( \frac{Ov(P)^{3/2} \cdot n}{V(P)^{1/2}} \right). \]

\[ \square \]

**Derivation for Odd Product Model**

To apply Newton-Raphson to optimizing the odd-product model (Section 3), we need the Jacobian matrix $J$ of derivatives of the degree errors with respect to the entries of $\ell$. Since $d$ does not vary with $\ell$, these derivatives are exactly $\frac{\partial d}{\partial \ell_i}$, which can be computed as:

\[ \frac{\partial d_i}{\partial \ell_j} = \frac{\partial}{\partial \ell_j} \sum_{k \in [n]} P_{ik} = \frac{\partial}{\partial \ell_j} \sum_{k \in [n]} \sigma(\ell_i + \ell_k) = \frac{\partial}{\partial \ell_j} \sigma(\ell_i + \ell_j) + \delta_{ij} \sum_{k \in [n]} \frac{\partial}{\partial \ell_j} \sigma(\ell_i + \ell_k) = \sigma(\ell_i + \ell_j) (1 - \sigma(\ell_i + \ell_j)) + \delta_{ij} \sum_{k \in [n]} \sigma(\ell_i + \ell_k) (1 - \sigma(\ell_i + \ell_k)) = P_{ij} (1 - P_{ij}) + \delta_{ij} \sum_{k \in [n]} P_{ik} (1 - P_{ik}). \]

**B \hspace{1em} Exact Embeddings in the CELL Model**

Recently, Rendsburg et al. [23] propose the CELL graph generator: a major simplification of the NetGAn algorithm for $[3]$, which gives comparable performance, much faster runtimes, and helps clarify the key components of the generator. CELL uses a simple low-rank factorization model. Here we prove that, when its rank parameter is $k$, the CELL model can ‘memorize’ any graph with degree bounded by $O(k)$. This allows the model to trivially produce distributions with very high expected triangle densities. However, as our main results show, this inherently requires memorization and high overlap.

Our result can be viewed as an extension of the results of [4], which considers a different edge independent model. The proof techniques are very similar. Interestingly, our result seem to indicate
that the good generalization of CELL in link prediction tests may mostly be due to the fact that this model is not fully optimized, to the point of memorizing the input.

The CELL Model. We first describe the CELL model introduced in [23].

1. Given a graph adjacency matrix \( A \in \{0, 1\}^{n \times n} \), let

\[
W^* = \min_{W \in \mathbb{R}^{n \times n}, \text{rank}(W) \leq k} \sum_{i,j=1}^{n} A_{ij} \log \sigma_{\text{rows}}(W)_{ij},
\]

where \( \sigma_{\text{rows}}(W) \) applies a softmax rowwise to \( W \) – ensuring that each row of \( \sigma_{\text{rows}}(W) \) sums to 1.

2. Let \( P^* = \sigma_{\text{rows}}(W^*) \) and let \( \pi \in \mathbb{R}^n \) be the eigenvector satisfying \( \pi^T P^* = \pi^T \).

3. Let \( P = \max(\text{diag}(\pi)P^*, (\text{diag}(\pi)P^*)^T) \).

4. Generate \( G \sim G(P) \).

Note that the last step described above is slightly different than the approach taken in CELL. Rather than use an edge-independent model as in Def. [1], they form \( G \) by sampling edges without replacement, with probability proportional to the entries in \( P \). They also ensure that at least one edge is sampled adjacent to every node. However, this distinction is minor.

Unconstrained Optimum. We first show that, if the rank constraint in (2) is removed, then the optimal \( W^* \) has \( \sigma_{\text{rows}}(W^*) = P^* = D^{-1}A \), where \( D \) is the diagonal degree matrix. At this minimum, we can check that \( \pi_i = d_i \), the degree of the \( i \)th node, and thus \( \text{diag}(\pi) = D \) and \( P = A \). That is, the model simply outputs the input graph with probability 1.

Theorem 8 (CELL Optimum). The unconstrained CELL objective function (2) is minimized when \( \sigma_{\text{rows}}(W) = D^{-1}A \). At this minimum, the edge independent model \( P \) is simply \( A \). That is, the model just returns the input graph with probability 1.

Proof. It suffices to consider the \( i \)th row of \( W \) for each \( i \in [n] \), since the objective function of (2) breaks down rowwise. Let \( w_i, a_i \in \mathbb{R}^n \) be the \( i \)th rows of \( \sigma_{\text{rows}}(W) \) and \( A \) respectively. Note that \( w_i \) is a probability vector, with \( w_i(j) \geq 0 \) for all \( j \) and \( \sum_{j=1}^{n} w_i(j) = 1 \).

We seek to minimize \( \sum_{j=1}^{n} A_{ij} \log[w_i(j)] \). We need to show that this objective is minimized when \( w_i = 1/d_i \cdot a_i \) – i.e., when \( w_i \) places mass \( 1/d_i \) at each nonzero entry in \( a_i \). Since \( \sum_{j=1}^{n} w^*(j) = 1 \), we must have \( w^*(j) \leq 1/d_i \) for some \( j \) where \( a_i = 1 \). In turn, there must be some \( j' \) with either (1) \( w^*(j') \geq 0 \) and \( a_i(j') = 0 \) or (2) \( w^*(j') \geq 1/d_i \) and \( a_i(j') = 1 \). In case (1), clearly moving \( w^*(j') \) mass from \( j' \) to \( j \) will decrease the objective function. In case (2), due to the concavity of the log function, moving \( w^*(j') - 1/d_i \) mass from \( j' \) to \( j \) will also decrease the objective function. Thus, \( w^* \) cannot be a minimizer, completing the proof.

Rank-Constrained Optimum. We next show that the unconstrained optimum of \( \sigma_{\text{rows}}(W) = D^{-1}A \), which leads to CELL memorizing the input graph (Thm. [8]) can be achieved even with the rank constraint of (2), as long as \( k \geq 2\Delta + 1 \), where \( \Delta \) is the maximum degree of the input graph.

Theorem 9 (CELL Exact Factorization). If \( A \) is an adjacency matrix with maximum degree \( \Delta \), there is a rank \( 2\Delta + 1 \) matrix \( W \) with

\[
\sigma_{\text{rows}}(W) = D^{-1}A + E
\]

where \( \|E\|_2 \leq \epsilon \). Note that the rank of \( W \) does not depend on \( \epsilon \), and so we can drive \( \epsilon \to 0 \) and find a rank-\( 2\Delta + 1 \) \( W \) which is arbitrarily close to minimizing (2) and thus produces \( P \) which is arbitrarily close to \( A \).

Proof. Let \( V \in \mathbb{R}^{n \times 2\Delta + 1} \) be the Vandermonde matrix with \( V_{i,j} = t^{j-1} \). For any \( x \in \mathbb{R}^{2\Delta + 1} \),

\[
[Vx](t) = \sum_{j=1}^{2\Delta + 1} x(j) \cdot t^{j-1}.
\]

That is: \( Vx \) is a degree \( 2\Delta \) polynomial evaluated at the integers \( t = 1, \ldots, n \).
Let $a_i$ be the $i^{th}$ row of $A$. Note that $a_i$ has at most $\Delta$ nonzeros whose positions we denote by $t_1, t_2, \ldots, t_d$. To prove the theorem, for each row $a_i$, we will construct a polynomial $Vx_i$, which has the same positive value at each $t_1, t_2, \ldots, t_d$ and is negative all other integers $t$. Then, we will let $X \in \mathbb{R}^{2\Delta + 1 \times n}$ be the matrix with columns $x_i$ and $W = (VX)^T$. Note that $\text{rank}(W) \leq 2\Delta + 1$, and $W$ is equal to a fixed positive value whenever $A$ is one and negative whenever it is zero. If we scale $W$ by a very large number (which does not affect its rank), we will have $\sigma_{\text{rows}}(W)$ arbitrarily close to $D^{-1}A$, since the rowwise softmax will place equal probability on each positive entry in row $i$ of $W$ and arbitrarily close to 0 probability on each negative. So the row will exactly have $d_i$ at the nonzero entries of $a_i$, entries each equal to $\frac{1}{d_i}$.

It remains to exhibit the polynomial need to construct $W$. We start by constructing a polynomial of degree $2\Delta$ that is positive on each nonzero position $t_1, t_2, \ldots, t_d$ of $a_i$ and negative at all other indices. Later we will modify this polynomial to have the same positive value at each nonzero position of $a_i$. Let $r_{j,L}$ and $r_{j,U}$ be any values with $t_j - 1 < r_{j,L} < t_j$ and $t_j < r_{j,U} < t_j + 1$. Consider the polynomial with roots at each $r_{j,L}$ and $r_{j,U}$. This polynomial has $2d_i \leq 2\Delta$ roots and so degree at most $2\Delta$. It will flip signs just at each $r_{j,L}$ and $r_{j,U}$, and will in fact have the same sign at $t_1, t_2, \ldots, t_d$ (either positive or negative). Simply negating the coefficients we can ensure that this sign is positive, while it is negative at all other indices, giving the result.

The polynomial above can be written as $p(t) = \prod_{j=1}^{d_i} (t - r_{j,U})(t - r_{j,L})$. Choose $r_{j,U} = t_j + \epsilon w_j$ and $r_{j,L} = t_j - \epsilon w_j$, where $\epsilon$ is arbitrarily small and $w_j$ is a weight chosen specifically for $t_j$ which we’ll set later. We have for any $k = 1, \ldots, d_i$,

$$\lim_{\epsilon \to 0} \frac{p(t_k)}{\epsilon^2} = \lim_{\epsilon \to 0} \frac{\prod_{j=1}^{\Delta}(t_k - t_j + \epsilon w_j)(t_k - t_j - \epsilon w_j)}{\epsilon^2} = \lim_{\epsilon \to 0} \frac{-\epsilon^2 w_k^2 \cdot \prod_{j \neq k}(t_k - t_j)^2}{\epsilon^2} = -w_k^2 \cdot \prod_{j \neq k}(t_k - t_j)^2.$$

This, if we set $w_k = \frac{1}{\prod_{j \neq k}(t_k - t_j)}$, in the limit as $\epsilon \to 0$ we will have $p(t_k)/\epsilon^2 = -1$. If we negate and scale the polynomially appropriately (which doesn’t change its degree) we will have $p(t_k)$ arbitrarily close to one for each nonzero index $t_k$, and negative for each zero index. This gives the theorem. \qed

**C Omitted Experimental Results**

We now include descriptions and plots of metrics for 5 other networks. The basic statistics of each network are listed in Table I. We treat all networks as binary, in that we set all non-zero weights to 1, and undirected, in that if edge $(i, j)$ appears in the network, we also include edge $(j, i)$. See Figures 1, 4, 5, 6, and 7 for the plots.

1. **FACEBOOK**: A union of ego networks of Facebook users.
2. **CORA**: A collection of scientific publications and the citations among them.
3. **ROAD-MINNESOTA**: A road network from the state of Minnesota. Each intersection is a node.
4. **PPI**: A subgraph of the PPI network for Homo Sapiens. Vertices represent proteins and edges represent interactions.
Figure 4: Metrics for CORA.

Figure 5: Metrics for ROAD-MINNESOTA.
D Code for Reproducing Results

Code is available at [https://github.com/konsotirop/edge_independent_models](https://github.com/konsotirop/edge_independent_models). Our implementation of the methods we introduce is written in Python and uses the NumPy [34] and SciPy [36] packages. Additionally, to calculate the various graph metrics, we use the following packages: powerlaw [33] and MACE (MAximal Clique Enumerator) [35].
References


