Testing Triangle-Freeness in General Graphs

Noga Alon * Tali Kaufman † Michael Krivelevich ‡ Dana Ron§

Abstract

In this paper we consider the problem of testing whether a graph is triangle-free, and more generally, whether it is H-free, for a fixed subgraph H. The algorithm should accept graphs that are triangle-free and reject graphs that are far from being triangle-free in the sense that a constant fraction of the edges should be removed in order to obtain a triangle-free graph. The algorithm is allowed a small probability of error.

This problem has been studied quite extensively in the past, but the focus was on dense graphs, that is, when $d = \Theta(n)$, where d is the average degree in the graph and n is the number of vertices. Here we study the complexity of the problem in general graphs, that is, for varying d. In this model a testing algorithm is allowed to ask neighbor queries (i.e., "what is the i-th neighbor of vertex v"), vertex-pair queries (i.e., "is there an edge between vertices v and u"), and degree queries (i.e., "what is the degree of vertex v").

Our main finding is a lower bound of $\Omega(n^{1/3})$ on the necessary number of queries that holds for *every* $d < n^{1-\nu(n)}$, where $\nu(n) = o(1)$. Since when $d = \Theta(n)$ the number of queries sufficient for testing has been known to be independent of n, we observe an abrupt, *threshold-like* behavior of the complexity of testing around n. This lower bound holds for testing H-freeness of every non-bipartite subgraph H.

Additionally, we provide sub-linear upper bounds for testing triangle-freeness that are at most quadratic in the stated lower bounds, and we describe a transformation from certain one-sided error lower bounds for testing subgraph-freeness to two-sided error lower bounds.

Finally, in the course of our analysis we show that dense random Cayley graphs behave like quasi-random graphs in the sense that relatively large subsets of vertices have the "correct" edge density. The result for subsets of this size cannot be obtained from the known spectral techniques that only supply such estimates for much larger subsets.

*Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel, and Institute for Advanced Study, Princeton, NJ 08540, USA. E-mail: nogaa@post.tau.ac.il. Research supported in part by a USA Israel BSF grant, by a grant from the Israel Science Foundation and by the Von Neumann Fund.

†Computer Science and Artificial Intelligence Lab, MIT, Cambridge, MA, 02138, USA. E-mail: kaufmant@mit.edu. This work is part of the author's Ph.D. thesis done at Tel Aviv University under the supervision of Prof. Noga Alon, and Prof. Michael Krivelevich. Research done in part while the author visited the Radcliffe Institute for Advanced Study, Harvard University.

‡Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: krivelev@post.tau.ac.il. Research supported in part by a USA Israeli BSF grant and by a grant from the Israel Science Foundation.

§Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: danar@eng.tau.ac.il. Research done in part while the author was a fellow at the Radcliffe Institute for Advanced Study, Harvard University. Research supported in part by a grant from the Israel Science Foundation.

1 Introduction

Property Testing is the study of the following type of computational tasks. Let P be some predetermined property, where we shall be interested in properties of combinatorial nature. The task is to distinguish quickly and reliably between input objects possessing property P and input objects that are "far" from having P, where distance is measured in some appropriately defined quantitative sense. In order to fulfill this task, the testing algorithm is given *query access* to a description of the input object O, adhering to some pre-agreed upon format; the algorithm's complexity is measured by the number of queries it asks before reaching a reliable decision. A testing algorithm is expected to query a sublinear portion of the input; hence, randomness plays an essential role in developing property testing algorithms. Property testing was defined by Rubinfeld and Sudan [RS96], who focused on testing algebraic properties. Goldreich, Goldwasser and Ron [GGR98] initiated the study of combinatorial property testing, and in particular considered testing properties of graphs. Since then, the field has progressed enormously, with many papers devoted to it, where one of the main focuses is on testing graph properties. An interested reader is invited to consult the surveys [Fis01, Ron01] for more details.

In this work we consider the problem of testing subgraph-freeness, and in particular triangle-freeness, in general graphs. Let n denote the number of vertices in the graph, let d denote the average degree, and let $d_{\max}$ denote the maximum degree. Given a distance parameter $\epsilon > 0$, we would like to design an algorithm that distinguishes with high probability between the case that the graph contains no triangles and the case in which more than $\epsilon \cdot dn$ edges should be removed so that no triangles remain. To this end we allow the algorithm query access to the graph. In particular, for any vertex of its choice, the algorithm may ask for the degree of the vertex, it may ask to get the i-th neighbor of the vertex for every $i \leq n$ (if the vertex has less than i neighbors then a null answer is returned), and it may ask whether there is an edge between any two vertices.[1]

Subgraph-freeness, and more specifically, triangle-freeness, is one of the most basic problems studied in property testing. The interest in this problem is both due to the fact that triangle-freeness is a fundamental and simple graph property, and it is due to the relation between triangle-freeness and the study of dense sets of integers with no three-term arithmetic progression.

Dense graphs. Most of the focus in previous works was on testing triangle-freeness in dense graphs, that is, when $d = \Theta(n)$. For this class of graphs the most appropriate input graph representation is the graph adjacency matrix, and a testing algorithm is allowed to query whether (u, v) is an edge of an input graph G, where $u, v \in V(G)$ (the so called *vertex-pair* queries). The authors of [AFKS00] showed that it is possible to test triangle-freeness in dense graphs using a number of queries that is *independent* of n, and has a tower-type behavior in $1/\epsilon$. Alon [Alo02] proved that a super-polynomial dependence on $1/\epsilon$ is necessary for testing subgraph-freeness of all non-bipartite subgraphs. When the fixed subgraph is bipartite then $O(1/\epsilon)$ queries suffice [Alo02]. It is also observed in [Alo02] (and much earlier, though implicitly, in [RS76]) that the problem of testing triangle-freeness is intimately related to the famous (and very hard) problem of the existence of dense sets of integers without a three-term arithmetic progression. Alon's lower bound, which

[1] Clearly, a degree query to a vertex v can be implemented by performing at most $\log n$ queries concerning the neighbors of v. Therefore, an algorithm can do without degrees queries and incur at most a logarithmic factor overhead. For simplicity of the presentation, we allow the algorithm to perform these queries.

was proved for one-sided error algorithms, was extended in [AS04b] to two-sided error algorithms. Other related results include [AS04a].

Bounded-Degree graphs. In the other extreme, an input graph is assumed to have its maximum degree bounded by an absolute constant $d_{\max} = O(1)$. In such a case, the input graph is usually represented by an array of incidence lists of all of its vertices (of length at most $d_{\max}$ each); accordingly, a testing algorithm queries the i-th neighbor of a vertex v, where $1 \leq i \leq d_{\max}$ (the so called *neighbor* queries). As was observed in [GR02], in this case $O(1/\epsilon)$ queries suffice for testing triangle-freeness. More generally, $O(d^{\tau}/\epsilon)$ queries suffice for testing H-freeness in graphs with maximum degree $O(d)$, where τ is the diameter of H.

General graphs. In this work we study the complexity of testing triangle-freeness of graphs that lie between the two extremes. Namely, we would like to understand the dependence of the query complexity on the average degree d, and we do not want to necessarily assume that $d_{\max} = O(d)$. In the latter aspect we follow the work [PR02] on testing the diameter of sparse, but unbounded-degree, graphs, and in both aspects we follow the work [KKR04] on testing bipartiteness of general graphs. In particular, as in [KKR04] we allow queries of both types – vertex-pair queries and neighbor queries, as well as degree queries (i.e., the algorithm may query the degree of any vertex). Note that the fact that the graph has varying degrees makes the task of testing triangle-freeness significantly harder. Consider for example sparse graphs, that is, graphs with average degree $d = O(1)$. As we mentioned before, when $d_{\max} = O(1)$, $O(1/\epsilon)$ queries suffice for testing triangle-freeness. However, our work shows that when $d_{\max} = \Theta(n)$ and $d = O(1)$, the number of queries required for testing triangle-freeness is $\Omega(\sqrt{n})$.

Our contributions. The main contributions of this paper, on a qualitative level, are as follows:

- We discover a threshold-type behavior in testing triangle-freeness: whenever $d = O(n^{1-\nu(n)})$, where $\nu(n)$ is a function that satisfies $\nu(n) = o(1)$, the number of queries that are necessary to test triangle-freeness is $\Omega(n^{1/3})$, while, as discussed above, for $d = \Theta(n)$ the query complexity is a function of ϵ only. This is in sharp contrast with the results of [KKR04], where a smooth behavior of the complexity of testing bipartiteness as a function of d was described;
- We provide a transformation from lower bounds for testing triangle-freeness using one-sided error algorithms to those for two-sided error algorithms; though the suggested transformation is stated and proved for triangles and carries some technical restrictions, it is general enough to capture a variety of lower bounds of this sort;
- We give quantitative lower and upper bounds for testing triangle-freeness in general graphs;
- We show that the edge distribution in random Cayley graphs is close to that of truly random graphs of the same edge density. This is proven by direct combinatorial and probabilistic arguments, without relying on the eigenvalue machinery, which is incapable of proving such results for subsets that are too small. Although we need this result for property testing purposes, we feel it is of enough independent interest to be stated here.

1.1 A lower bound and a sharp threshold

Our main result is:

Theorem 1 *There is a lower bound of $\Omega(n^{1/3})$ for testing triangle-freeness in general graphs. The lower bound holds for algorithms that are allowed two-sided error, and for every d that is upper bounded by $O(n^{1-\nu(n)})$ where $\nu(n) = \frac{\log\log\log n+4}{\log\log n}$. For some values of d the lower bound reaches $\Omega(n^{1/2})$.*

Theorem 1 is actually the union of three lower bounds (whose one-sided error versions are stated in Lemmas 1, 2 and 6), which are applied to different values of d. The exact expression for the lower bound is

$$\Omega\Big(\max\Big\{\sqrt{n/d}, \min\{d, n/d\}, \min\{\sqrt{d}, n^{2/3}/d^{1/3}\}\cdot n^{-o(1)}\Big\}\Big) \tag{1}$$

For a schematic illustration see Figure 1.

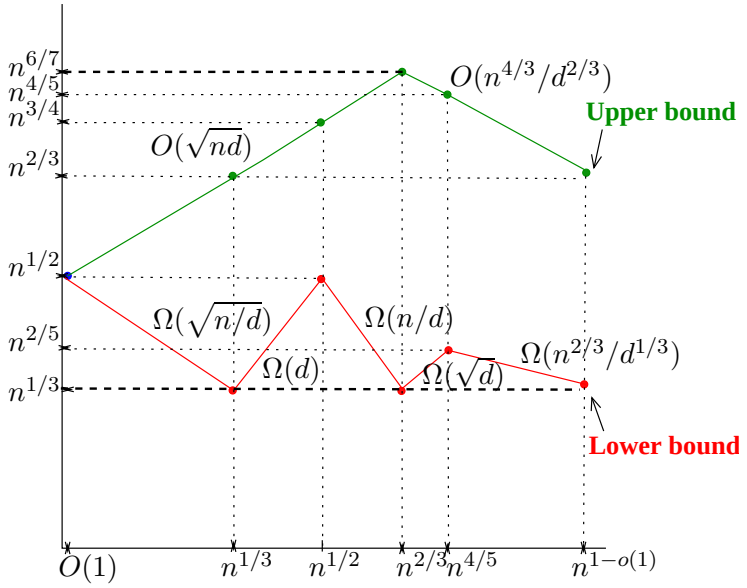

Figure 1: A schematic illustration of our lower and upper bounds. The x-axis represents the average degree d and the y-axis represents the bounds. For the sake of simplicity we ignore logarithmic factors in the bounds. Notice that the lower bound lies entirely above the horizontal line at height $n^{1/3}$, and the upper bound lies entirely below the horizontal line at height $n^{6/7}$.

Recall that when $d = \Theta(n)$ then testing can be performed using a number of queries that is independent of n [AFKS00]. Thus we observe a sharp transition between our lower bound of $\Omega(n^{1/3})$ that holds until $d = n^{1-\nu(n)}$ (recall that $\nu(n) = o(1)$), and the query complexity at $d = \Theta(n)$, which does not depend on n. The abrupt change of testing complexity around linear density can be partially explained by the fact that the celebrated Regularity Lemma of Szemerédi (whose relevance to testing triangle-freeness has been first indicated in [RS76] and was made explicit in [AFKS00]) starts yielding meaningful results for graphs whose number of edges is almost quadratic

in the number of vertices. The exact behavior of the complexity of testing triangle-freeness when $n^{1-\nu(n)} \leq d \leq n$ remains open.

Remark. Using techniques that were previously applied in [Alo02] it is possible to extend Theorem 1 to testing subgraph-freeness for other non-bipartite subgraphs and to show that for any fixed non-bipartite graph H there is a constant $a = a(H) > 0$ such that an algorithm for testing H-freeness in graphs on n vertices with average degree $d \leq n^{1-o(1)}$ has to ask at least n^a queries. This lower bound holds for two-sided error algorithms, using both vertex-pair and neighbor queries. We wish to note that the difference between the complexity of testing subgraph-freeness of bipartite and non-bipartite subgraphs is caused by the difference in the behavior of their Turán numbers – they are subquadratic for the former and quadratic for the latter. A more detailed discussion can be found in [Alo02]. The technical details of the aforementioned general statement concerning testing H-freeness and its proof are quite cumbersome, while the threshold-type behavior of the query complexity as a function of average degree d is observed already for the case when H is a triangle. Therefore we prefer to concentrate on the basic case of testing triangle-freeness.

1.2 Upper bounds

We show that for every graph density, there exists an algorithm for testing triangle-freeness whose query complexity is sublinear in n. Furthermore, the upper bound is always at most quadratic in the corresponding lower bound (up to a factor of $\log n$).

Theorem 2 *There is an upper bound of $\tilde{O}(n^{6/7})$ for testing triangle-freeness in general graphs for every value of d. The upper bound can go down to $\tilde{O}(n^{1/2})$ for some values of d. In all cases, up to logarithmic factors, the upper bound is at most quadratic in the lower bound that holds for that density. If $d_{\max} = O(d)$ then the upper bound is $O(n^{4/5})$ for all values of d.*

The exact expression for our upper bound is $\tilde{O}\left(\min\left\{\sqrt{nd}/\epsilon^{3/2}, (n^{4/3}/d^{2/3})/\epsilon^2\right\}\right)$, where in the case that $d_{\max} = O(d)$, the first term is replaced by d/ϵ. For a schematic illustration, see Figure 1.

1.2.1 Tight results.

There are two cases in which our lower and upper bounds essentially match. The first case is graphs in which $d_{\max} = O(d)$ and $d \leq \sqrt{n}$. For this case the complexity is $\Theta(d)$ (for constant ϵ). The second case is general sparse graphs, that is, graphs for which $d = \Theta(1)$. For these graphs the complexity is $\tilde{\Theta}(\sqrt{n})$.

1.3 Our techniques

Behrend Graphs and Cayley graphs. In the proof of our third lower bound (Lemma 6), we build on graphs that are known as Behrend graphs, which were previously used in the context of testing triangle-freeness in [Alo02]. Here we prove that random Behrend graphs have a certain property that we can exploit in order to obtain our lower bound. Behrend graphs are variants of the well studied Cayley graphs, and our proof concerning properties of random Behrend graphs extends to Cayley graphs.

Specifically, we show that for dense random Cayley graphs the edge density in relatively large induced subgraphs is close to the edge density of the whole graph. It was previously shown [AR94] that random Cayley graphs are expanders and hence have the property that the density of every induced subgraph on sufficiently many vertices is very close to the density of the graph. However, the known techniques for proving this property are based on estimating the second eigenvalue of the graph's adjacency matrix, and do not supply any informative bounds for sets of vertices that are much smaller than the number of vertices divided by the square root of the degree. Our results for Cayley graphs apply both for Cayley graphs over Abelian and non-Abelian groups, while Behrend graphs were considered only in an Abelian setting. Our techniques are somewhat reminiscent of those of [AO95, Gre05].

A reduction from one-sided error lower bounds to two-sided error lower bounds. We obtain our two main lower bounds by first establishing lower bounds that hold for one-sided error algorithms. We then prove a transformation from one-sided error lower bounds to two-sided error lower bounds that holds under certain assumptions, and apply it to obtain our two-sided error lower bounds. This transformation may be of use in future lower bound proofs for subgraph freeness. We note that in [AS04b] a transformation was given in the case of dense graphs, but it is not applicable in general.

1.4 Subsequent work

In [Gug06] and [Ras06] our lower bound and upper bound, respectively, were improved somewhat for particular ranges of d. Specifically, Gugelmann [Gug06] slighly modifies one of our constructions, and obtains a lower bound of $\Omega(\min\{(nd)^{1/3}, n/d\})$, instead of our lower bound of $\Omega(\min\{d, n/d\})$. Given our additional lower bound of $\Omega(\sqrt{n/d})$, this improves on our result for d that ranges between $\Omega(n^{1/5})$ and $O(n^{1/2})$.

In terms of upper bounds, Rast [Ras06] describes an algorithm that combines ideas from our algorithms in a non-trivial manner, and obtains an algorithm whose query complexity (in terms of its dependence on n and d) is $O\left(\max\left\{(nd)^{4/9}, n^{2/3}/d^{1/3}\right\}\right)$. This improves on our result for d that ranges between $\Omega(n^{2/5})$ and $O(n^{4/5})$.

1.5 Paper organization

After giving some preliminary definitions in Section 2, we proceed to lower bounds in Sections 3, 4, 5. In particular, Section 5 describes Behrend graphs and proves a result on the edge density of random Behrend graphs, whose more general version (applicable to random Cayley graphs of Abelian groups) is discussed in Section 8. Section 6 describes a reduction from lower bounds for two-sided testers to those for one-sided testers. Section 7 contains upper bounds for testing triangle-freeness.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with n vertices labeled $1, \ldots, n$, and let d denote the average degree in G, where we assume that $d = \Omega(1)$.[2] For each vertex $v \in V$ let $\deg(v)$ denote the degree of vertex v. The edges incident to v (and their end-points, the neighbors of v), are labeled from 1 to $\deg(v)$. Note that each edge has two, possibly different, labels, one with respect to each of its end-points. For a graph G and a subset of vertices $U \subseteq V$, we refer to the edges in the subgraph of G that is induced by U as the edges *spanned* by U in G.

A graph G is said to be *triangle-free* if for every three vertices u, v, w in G, at least one of the three vertex-pairs (u, v), (v, w), or (w, u) is not an edge in G. A graph G is *ϵ-far from (being) triangle-free* if it is necessary to remove more than ϵdn edges from G in order to obtain a triangle-free graph. We note that since the number of edges in the graph is $(dn)/2$, the standard definition of ϵ-far would be that more than $(\epsilon dn)/2$ edges should be removed so that the graph becomes triangle-free. In order to simplify the presentation we slightly modify the definition.

A testing algorithm for triangle-freeness is required to accept with probability at least 2/3 every graph that is triangle-free and to reject with probability at least 2/3 every graph that is ϵ-far from being triangle-free, where ϵ is a given distance parameter. If the algorithm always accepts triangle-free graphs then it has *one-sided error*, otherwise it has *two-sided error*. In order to perform this task the testing algorithm is allowed the following types of queries:

- *Degree* queries: for any vertex u of its choice, the algorithm can obtain $\deg(u)$.
- *Neighbor* queries: for any vertex u and index $1 \leq i \leq \deg(u)$, the algorithm may obtain the i-th neighbor of vertex u.
- *Vertex-pair* queries: for any pair of vertices (u, v), the algorithm can query whether there is an edge between u and v in G.

3 A Lower Bound of $\Omega\left(\sqrt{n/d}\right)$

In this section we establish our first, and simplest lower bound.

Lemma 1 *Every algorithm for testing triangle-freeness must perform $\Omega(\sqrt{n/d})$ queries. This lower bound holds for two-sided error algorithms as well.*

Proof: In order to prove a two-sided error lower bound of $\Omega(q)$ queries for testing triangle-freeness, it suffices to describe two families of graphs for which the following two conditions hold. (1) The graphs in the first family are all triangle-free, while the graphs in the second family are all $\Theta(1)$-far from being triangle-free. (2) Any algorithm that distinguishes with constant probability between a graph selected uniformly in one family, and a graph selected uniformly in the second family, must perform $\Omega(q)$ queries.

[2] Our results can be extended to the case that $d = o(1)$ (that is, very sparse graphs). However, for the sake of simplicity, and since we believe that the very sparse case is of less interest, we assume that $d = \Omega(1)$.

In particular, consider the following two families of graphs over n vertices and with average degree d. Each family is determined by a single graph, and consists of all possible $n!$ labelings of the vertices of the graph. Hence it suffices to describe the two graphs (one per family). In one graph there is a complete bipartite graph between two sets of vertices, each of size $\sqrt{nd/2}$. In the other graph there is a clique of size $\sqrt{nd}$. (We ignore rounding issues here.) In addition, in both graphs the remaining vertices are isolated. The first graph is clearly triangle-free and it is not hard to verify that the second graph is $\Theta(1)$-far from being triangle-free. However, in order to distinguish between the two graphs (or more precisely, in order to distinguish between graphs that are selected uniformly from each of the two families), the algorithm must obtain a vertex in the complete bipartite subgraph / clique. To this end the algorithm must perform $\Omega(n/\sqrt{nd}) = \Omega(\sqrt{n/d})$ queries. ■

4 A Lower Bound of $\Omega(\min\{d, n/d\})$

Our next lower bound improves on the lower bound in Section 3 when $d > n^{1/3}$.

Lemma 2 *Every one-sided error algorithm for testing triangle-freeness must perform $\Omega(\min\{d, n/d\})$ queries. This lower bound holds even when $d_{\max} = O(d)$.*

The lower bound in Lemma 2 is extended to two-sided error algorithms in Section 6.

We prove Lemma 2 by describing a distribution on graphs such that the following holds: On one hand almost all of the support of the distribution is on graphs that are far from triangle-free. On the other hand, every algorithm that uses neighbor and/or vertex-pair queries, must perform $\Omega(\min\{d, n/d\})$ queries before it views a triangle (with sufficiently high constant probability). Since we currently focus on testing algorithms that have one-sided error, this implies a lower bound on the query complexity of such algorithms.

We start by considering the case that $d = c \cdot \sqrt{n}$ for a particular constant $c < 1$. We later discuss how to deal with the case that $d > c \cdot \sqrt{n}$ and with the case that $d < c \cdot \sqrt{n}$.

4.1 Definition of the lower-bound distribution

The graphs we consider below are d-regular for $d = \Theta(\sqrt{n})$, but may have multiple edges. At the end of this section we discuss how to remove the multiple edges. Let D_Δ be a distribution over graphs with n vertices and degree $d = \frac{2}{3}\sqrt{n/3}$ that is defined as follows. A graph is generated by first partitioning the vertices into equal-size subsets of size $n' = n/3$ denoted V_1, V_2, V_3. Next, between each pair of subsets, $d' = d/2 = \sqrt{n'}/3$ random perfect matching are selected. In all that follows we assume that n' is sufficiently large ($n' > 100$ suffices).

Lemma 3 *With probability $1 - o(1)$, a graph chosen uniformly according to the distribution D_Δ is $\Omega(1)$-far from being triangle-free.*

Proof: We shall show that with high probability over the choice of a graph according to D_Δ, there are at least $\gamma \cdot nd$ edge-disjoint triangles in the graph, for some constant γ.

Let the vertices in each set V_ℓ, $\ell \in \{1,2,3\}$, be denoted $v_{\ell,1}, \ldots, v_{\ell,n'}$. For each pair of vertices $v_{\ell,i} \in V_\ell$, $v_{\ell',j} \in V_{\ell'}$ where $\ell \neq \ell'$, let $\alpha_{i,j}^{\ell,\ell'}$ be a 0/1 random variable that is 1 if there is an edge between $v_{\ell,i}$ and $v_{\ell,j}$ and is 0 otherwise. Then

$$\Pr\left[\alpha_{i,j}^{\ell,\ell'} = 1\right] = 1 - \left(1 - \frac{1}{n'}\right)^{d'} \tag{2}$$

and so

$$\Pr\left[\alpha_{i,j}^{\ell,\ell'} = 1\right] \leq \frac{d'}{n'} \tag{3}$$

and

$$\Pr\left[\alpha_{i,j}^{\ell,\ell'} = 1\right] \geq \frac{d'}{n'} - \binom{d'}{2} \cdot \frac{1}{(n')^2} \geq \frac{d'}{n'} - \frac{1}{18n'} = \frac{d'}{n'} \cdot \left(1 - \frac{1}{18d'}\right) \tag{4}$$

Next consider any choice of three vertices $v_{1,i} \in V_1$, $v_{2,j} \in V_2$ and $v_{3,k} \in V_3$. Let $\Delta_{i,j,k}$ denote the 0/1 random variable that is 1 if and only if there is a triangle between the three vertices. Since the choice of the matchings between V_1 and V_2 is independent of the choice of the matchings between V_2 and V_3 and the choice of the matching between V_1 and V_3,

$$\Pr[\Delta_{i,j,k} = 1] \leq \left(\frac{d'}{n'}\right)^3 \tag{5}$$

and

$$\Pr[\Delta_{i,j,k} = 1] \geq \left(\frac{d' - 1/18}{n'}\right)^3 = \left(\frac{d'}{n'}\right)^3 \cdot \left(1 - \frac{1}{18d'}\right)^3 \tag{6}$$

Let $\beta_{i,j,k}$ denote a 0/1 random variable that is 1 if and only if there is a triangle *different from* $(v_{1,i}, v_{2,j}, v_{3,k})$ that includes one of the edges $(v_{1,i}, v_{2,j})$ or $(v_{1,i}, v_{3,k})$ or $(v_{2,j}, v_{3,k})$. That is, there is a triangle of the form $(v_{1,i'}, v_{2,j}, v_{3,k})$, where $i' \neq i$, or $(v_{1,i}, v_{2,j'}, v_{3,k})$ where $j' \neq j$, or $(v_{1,i}, v_{2,j}, v_{3,k'})$ where $k' \neq k$. Since the probability of having an edge $(v_{\ell,r}, v_{\ell',s})$ conditioned on having the edge $(v_{\ell,r}, v_{\ell',s'})$, for any choice of $s' \neq s$, is at most $1 - \left(1 - \frac{1}{n'}\right)^{d'-1}$,

$$\begin{aligned}
\Pr[\Delta_{i,j,k} = 1 \;\&\; \beta_{i,j,k} = 1] &= \Pr[\Delta_{i,j,k} = 1] \cdot \Pr[\beta_{i,j,k} = 1 \mid \Delta_{i,j,k} = 1] &(7)\\
&\leq \left(\frac{d'}{n'}\right)^3 \cdot 3 \cdot (n' - 1) \cdot \left(\frac{d' - 1}{n'}\right)^2 \leq \left(\frac{d'}{n'}\right)^3 \cdot \frac{3(d'-1)^2}{n'} &(8)
\end{aligned}$$

It follows that

$$\begin{aligned}
&\Pr[\Delta_{i,j,k} = 1 \;\&\; \beta_{i,j,k} = 0]\\
&= \Pr[\Delta_{i,j,k} = 1] - \Pr[\Delta_{i,j,k} = 1 \;\&\; \beta_{i,j,k} = 1] &(9)\\
&\geq \left(\frac{d' - 1/18}{n'}\right)^3 - \left(\frac{d'}{n'}\right)^3 \cdot \frac{3(d'-1)^2}{n'} &(10)\\
&= \left(\frac{d'}{n'}\right)^3 \cdot \left(\left(1 - \frac{1}{18d'}\right)^3 - 3 \cdot \left(1 - \frac{1}{d'}\right)^2 \cdot \frac{(d')^2}{n'}\right) &(11)\\
&\geq \frac{1}{2} \cdot \left(\frac{d'}{n'}\right)^3 = \frac{1}{54} \cdot (n')^{-3/2} &(12)
\end{aligned}$$

where we have used our assumption that n' and $d' = \sqrt{n'}/3$ are sufficiently large. Hence, the expected number of triples $(v_{1,i}, v_{2,j}, v_{3,k})$ that constitute a triangle that does not share an edge with any other triangle is at least

$$(n')^3 \cdot \frac{1}{54} \cdot (n')^{-3/2} = \frac{1}{54} \cdot (n')^{3/2} = \frac{1}{54} \cdot n \cdot d \tag{13}$$

Next, we shall use Chebyshev's inequality to show that with high probability, the number of such triples is not much smaller than this expected value.

For each triple $v_{1,i}, v_{2,j}, v_{3,k}$ let $\eta_{i,j,k}$ be the 0/1 random variable that is 1 if and only if $(v_{1,i}, v_{2,j}, v_{3,k})$ constitute a triangle, and there is no other triangle that shares any of its edges. Using our previous notation we have that

$$\Pr[\eta_{i,j,k} = 1] = \Pr[\Delta_{i,j,k} = 1 \ \& \ \beta_{i,j,k} = 0] \tag{14}$$

which implies (using Equations (9)–(13)) that

$$\left(\mathrm{Exp}\left[\sum_{i,j,k} \eta_{i,j,k} \right] \right)^2 \geq (n')^3/c \tag{15}$$

for some constant $c > 1$. By Chebyshev's inequality,

$$\Pr\left[\sum_{i,j,k} \eta_{i,j,k} < \frac{1}{2}\mathrm{Exp}\left[\sum_{i,j,k} \eta_{i,j,k} \right] \right] \leq \frac{4\mathrm{Var}\left[\sum_{i,j,k} \eta_{i,j,k}\right]}{\left(\mathrm{Exp}\left[\sum_{i,j,k} \eta_{i,j,k}\right]\right)^2} < \frac{c'\mathrm{Var}\left[\sum_{i,j,k} \eta_{i,j,k}\right]}{(n')^3} \tag{16}$$

for a constant $c' > 1$. We would like to bound the variance of $\sum_{i,j,k} \eta_{i,j,k}$:

$$\mathrm{Var}\left[\sum_{i,j,k} \eta_{i,j,k} \right] = \mathrm{Exp}\left[\left(\sum_{i,j,k} \eta_{i,j,k} \right)^2 \right] - \left(\mathrm{Exp}\left[\sum_{i,j,k} \eta_{i,j,k} \right] \right)^2 \tag{17}$$

Now,

$$\begin{aligned} \mathrm{Exp}\left[\left(\sum_{i,j,k} \eta_{i,j,k} \right)^2 \right] &= \sum_{i,j,k} \sum_{i',j',k'} \mathrm{Exp}[\eta_{i,j,k} \cdot \eta_{i',j',k'}] \qquad (18) \\ &= \sum_{i \neq i'} \sum_{j \neq j'} \sum_{k \neq k'} \mathrm{Exp}[\eta_{i,j,k} \cdot \eta_{i',j',k'}] \\ &\quad + 3 \cdot \sum_{i} \sum_{j \neq j'} \sum_{k \neq k'} \mathrm{Exp}[\eta_{i,j,k} \cdot \eta_{i,j',k'}] \\ &\quad + 3 \cdot \sum_{i,j} \sum_{k \neq k'} \mathrm{Exp}[\eta_{i,j,k} \cdot \eta_{i,j,k'}] \\ &\quad + \sum_{i,j,k} \mathrm{Exp}[\eta_{i,j,k}^2] \qquad (19) \end{aligned}$$

while

$$\left(\mathrm{Exp}\left[\sum_{i,j,k}\eta_{i,j,k}\right]\right)^2 = \sum_{i,j,k}\sum_{i',j',k'}\mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}] \quad (20)$$

$$> \sum_{i\neq i'}\sum_{j\neq j'}\sum_{k\neq k'}\mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}] \quad (21)$$

Therefore,

$$\begin{aligned}&\mathrm{Var}\left[\sum_{i,j,k}\eta_{i,j,k}\right]\\ &\leq \sum_{i\neq i'}\sum_{j\neq j'}\sum_{k\neq k'}\Big(\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i',j',k'}]-\mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}]\Big)\\ &\quad +3\cdot\sum_{i}\sum_{j\neq j'}\sum_{k\neq k'}\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i,j',k'}]+3\cdot\sum_{i,j}\sum_{k\neq k'}\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i,j,k'}]+\sum_{i,j,k}\mathrm{Exp}[\eta_{i,j,k}^2] \quad (22)\end{aligned}$$

First observe that by definition of $\eta_{i,j,k}$, for every i,j and $k\neq k'$, since if both $(v_{1,i},v_{2,j},v_{3,k})$ and $(v_{1,i},v_{2,j},v_{3,k'})$ are triangles then they share an edge, we get that

$$\eta_{i,j,k}\cdot\eta_{i,j,k'}=0\ . \quad (23)$$

Next observe that since $\Pr[\eta_{i,j,k}=1]\leq\Pr[\Delta_{i,j,k}=1]$, where by Equation (5) $\Pr[\Delta_{i,j,k}=1]\leq(d'/n')^3$,

$$\sum_{i,j,k}\mathrm{Exp}[\eta_{i,j,k}^2]=\sum_{i,j,k}\mathrm{Exp}[\eta_{i,j,k}]\leq(n')^3\cdot\left(\frac{d'}{n'}\right)^3=(d')^3 \quad (24)$$

We turn to the sum over triangles that share a (single) vertex. Conditioned on there being a triangle $(v_{1,i},v_{2,j},v_{3,k})$ i.e., $\Delta_{i,j,k}=1$, the probability of having the edge $(v_{1,i},v_{2,j'})$ for $j'\neq j$, i.e., $\eta_{i,j}^{1,2}=1$ (and similarly the edge $(v_{1,i},v_{3,k'})$ for $k'\neq k$) is upper bounded[3] by $1-\left(1-\frac{1}{n'}\right)^{d'-1}<\frac{d'}{n'}$. The probability of having the edge $(v_{2,j'},v_{3,k'})$ (conditioned on the existence of the triangle $(v_{1,i},v_{2,j},v_{3,k})$) is upper bounded by $1-\left(1-\frac{1}{n-1}\right)^{d'}$, which is at most $\frac{2d'}{n'}$ for $n'\geq 2$. Therefore,

$$\begin{aligned}\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i,j',k'}] &\leq \mathrm{Exp}[\Delta_{i,j,k}\cdot\Delta_{i,j',k'}] &(25)\\ &= \Pr[\Delta_{i,j,k}=1]\cdot\Pr[\Delta_{i,j',k'}=1\mid\Delta_{i,j,k}=1] &(26)\\ &\leq 2\left(\frac{d'}{n'}\right)^6 &(27)\end{aligned}$$

and so

$$\sum_{i}\sum_{j\neq j'}\sum_{k\neq k'}\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i,j',k'}]\leq(n')^5\cdot\frac{2(d')^6}{(n')^6}=\frac{2(d')^6}{n'} \quad (28)$$

[3]The reason this is an upper bound and equality does not necessarily hold is that the triangle $(v_{1,i},v_{2,j},v_{3,k})$ may have parallel edges. That is, the event $\Delta_{i,j,k}=1$ is a union of events in which each edge of the triangle may appear with different multiplicity.

To complete the proof we bound the difference between $\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i',j',k'}]$ and $\mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}]$ (for $i \neq i'$, $j \neq j'$, and $k \neq k'$). By definition,

$$\begin{aligned}&\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i',j',k'}] - \mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}]\\ &= \Pr[\eta_{i,j,k}=1]\cdot\left(\Pr[\eta_{i',j',k'}=1|\eta_{i,j,k}=1] - \Pr[\eta_{i',j',k'}=1]\right)\end{aligned} \tag{29}$$

Since the proof of the following claim is a bit technical, we give it in the appendix.

Claim 4

$$\Pr[\eta_{i',j',k'}=1 \mid \eta_{i,j,k}=1] = \left(1+O\left(\frac{1}{d'}\right)\right)\cdot\Pr[\eta_{i',j',k'}=1] \tag{30}$$

Since $\Pr[\eta_{i,j,k}=1] = \Pr[\eta_{i',j',k'}=1] \leq \left(\frac{d'}{n'}\right)^3$, Claim 4 implies that

$$\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i',j',k'}] - \mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}] = O\left(\frac{(d')^5}{(n')^6}\right) \tag{31}$$

Hence,

$$\sum_{i\neq i'}\sum_{j\neq j'}\sum_{k\neq k'}\mathrm{Exp}[\eta_{i,j,k}\cdot\eta_{i',j',k'}] - \mathrm{Exp}[\eta_{i,j,k}]\cdot\mathrm{Exp}[\eta_{i',j',k'}] = O\left((n')^6\cdot\frac{(d')^5}{(n')^6}\right) = O\left((d')^5\right) \tag{32}$$

By combining Equation (16) with Equations (22)–(32) we get that

$$\Pr\left[\sum_{i,j,k}\eta_{i,j,k} < \frac{1}{2}\mathrm{Exp}\left[\sum_{i,j,k}\eta_{i,j,k}\right]\right] = \frac{O((d')^5)+O((d')^3)}{(n')^3} = O\left(\frac{1}{d'}\right)\ . \tag{33}$$

Since we have shown that $\mathrm{Exp}\left[\sum_{i,j,k}\eta_{i,j,k}\right] = \Omega(nd)$, the lemma follows. ■

The case $d > \frac{2}{3\sqrt{3}}\sqrt{n}$. The distribution in this case is the same as described at the start of this section: the graph vertices are partitioned into three equal parts V_1, V_2, and V_3, and between every pair V_i and V_j we put $d/2$ random perfect matchings. We would like to show that with high probability the resulting graph contains at least $\frac{1}{c}\cdot nd$ edge-disjoint triangles in the graph, for some constant c. To this end we think of the matchings as being selected in $k = \frac{3\sqrt{3}}{2}\cdot\frac{d}{\sqrt{n}}$ rounds, where in each round $d' = \frac{1}{3\sqrt{3}}\cdot\sqrt{n}$ matching are selected between every pair V_i, V_j, $i \neq j$. For each round we can apply Lemma 3 and get that with high probability we have at least $\frac{1}{c}\cdot n\cdot d'$ edge-disjoint triangles. Observe that the triangles created in the different rounds are edge-disjoint. When $k = o(d')$ (i.e., $d = o(n)$) we can apply a union bound and get that with probability at least $1 - O(k/d') = 1 - o(1)$ we obtain $\frac{1}{c}\cdot n\cdot d$ edge-disjoint triangles. For larger k we can use the facts the the different rounds are independent. Therefore, with probability $1-\exp(-k) = 1-o(1)$ in at least $1/2$ of the rounds there are $\frac{1}{c}\cdot n\cdot d'$ edge-disjoint triangles, implying that there are at least $\frac{1}{2c}\cdot n\cdot d$ edge-disjoint triangles.

The case $d < \frac{2}{3\sqrt{3}}\sqrt{n}$. In this case we first partition the vertices into k parts $V^1, \ldots, V^k$ where $|V^i| = \frac{27}{4}d^2$. We then apply the construction described at the start of this section to each V^i. In this case too, for small k we can apply a union bound on the different V^i's, and once k is sufficiently large we can use the fact that the different subgraphs are constructed independently. In either case we get that with high probability, for at least a half of the V^i's there are $\Omega(|V^i| \cdot d)$ edge disjoint triangles within the subgraph induced by V^i.

4.2 The lower bound

Let A be any one-sided error algorithm for testing triangle-freeness, where A is allowed to perform both neighbor queries and vertex-pair queries. If A views a triangle in the tested graph then clearly it can reject the graph. However, since A is a one-sided error algorithm, if it terminates before viewing a triangle, then it must accept. Suppose we run A on a graph chosen according to D_Δ, with some (sufficiently small) constant ϵ. Since we have shown that with high probability such a graph is ϵ-far from being triangle free, the probability that A terminates before viewing a triangle must be small. Hence it remains to prove the following lemma.

Lemma 5 *Any algorithm whose goal is to detect, with high constant probability, a triangle in a graph selected according to D_Δ, must ask $\Omega(\min\{d, n/d\})$ queries.*

We note that the proof actually establishes the stronger statement: $\Omega(\min\{d, n/d\})$ queries are required to detect a cycle of any length.

Proof: We first present the argument for $d \geq \frac{2}{3\sqrt{3}}\sqrt{n}$ and later discuss the modifications required for $d < \frac{2}{3\sqrt{3}}\sqrt{n}$.

It will be convenient to view graphs in the support of D_Δ as being represented by "matchings over tables". Namely, there are 6 tables: $T_{1,2}, T_{1,3}, T_{2,1}, T_{2,3}, T_{3,1}, T_{3,2}$, two for each set of vertices V_b, $b \in \{1,2,3\}$. Each table $T_{b,b'}$ is of size $(n/3) \times (d/2)$: there is a row for each vertex v in V_b, and each entry in v's row corresponds to one of the $d/2$ edges that are incident to v and to vertices in $V_{b'}$. An edge between $u \in V_b$ and $v \in V_{b'}$ is represented by a pair of entries $(T_{b,b'}[u][i], T_{b',b}[v][j])$. Hence the $d/2$ perfect matchings between V_b and $V_{b'}$ correspond to a single perfect matching between the entries of the two tables $T_{b,b'}$ and $T_{b',b}$.

Let ALG be an algorithm that performs $Q = Q(n,d)$ queries and whose goal is to detect a triangle with probability at least 9/10. The probability is taken over the choice of the graph G in the support of D_Δ and the coin flips of the algorithm. Namely ALG is a (possibly probabilistic) mapping from *query-answer histories* $\langle (q_1, a_1), \ldots, (q_t, a_t) \rangle$, to q_{t+1} for every $t < Q$. A vertex-pair query is of the form $q_t = (u, v)$ where $u \in V_b$ and $v \in V_{b'}$ for some $b \neq b'$, $b, b' \in \{1,2,3\}$. The answer is either $a_t = (1, i, j)$, which denotes that there is an edge between u and v and it corresponds to the pair of entries $(T_{b,b'}[u][i], T_{b',b}[v][j])$, or $a_t = 0$, which denotes that there is no edge between u and v. A neighbor query is of the form $q_t = (u, b', i)$ where $u \in V_b$, $b' \neq b$. The answer is of the form $a_t = (v, j)$ where $v \in V_{b'}$ denoting that there is an edge between u and v and it corresponds to the pair of entries $(T_{b,b'}[u][i], T_{b',b}[v][j])$. We note that the mapping from query-answer histories to queries needs to be defined only on histories that are consistent with some graph in the support of D_Δ.

In what follows we define a process that answers the queries of the algorithm while generating a graph according to D_Δ. At any time t, the queries of the algorithm and the answers it is provided with determine the *knowledge graph* $G^t = (V^t, E^t, \overline{E}^t)$, where V^t are the vertices, E^t are the edges and $\overline{E}^t$ are the non-edges. Namely, V^t consists of all vertices that appeared in queries of the algorithm or in answers to neighbor queries. Similarly E^t consists of all pairs u, v such that either (u, v) was a vertex-pair query that was answered positively, or v was an answer to a neighbor query involving u, and $\overline{E}^t$ consists of all pairs u, v such that either (u, v) was a vertex-pair query that was answered negatively. For every edge $(u, v) \in E^t$ the knowledge graph will include the indices of the entries in the tables by which u and v are connected (that is, (i, j) such that $T_{b,b'}[u][i]$ is matched to $T_{b',b}[v][j]$.

Vertex-pair queries. Given a vertex-pair query $q_t = (u, v)$ where $u \in V_b$ and $v \in V_{b'}$, the process computes the probability, conditioned on the current knowledge graph, that (u, v) is an edge. Namely, it considers all graphs in the support of D_Δ that are consistent with G^{t-1} and answers positively with probability that is proportional to the number of these graphs in which there is an edge between u and v. Let $U_{t-1,b,b'}$ denote the number of unmatched entries in $T_{b,b'}$ at time t (before the t-th query). Note that this number equals the number of unmatched entries in $T_{b',b}$ at the same time.

The conditional probability that there is an edge between u and v is upper bounded by:

$$\frac{(d/2) \cdot (d/2)}{U_{t-1,b,b'} - |\overline{E}^{t-1}| \cdot (d/2) - (d/2)} \tag{34}$$

To verify this consider an iterative process in which the (at most $d/2$) unmatched entries in u's row in $T_{b,b'}$ are matched one by one to unmatched entries in $T_{b',b}$. After $0 \leq i < d/2$ steps, the probability of selecting any of the (at most $d/2$) unmatched entries in v's row in $T_{b',b}$ is the current number of unmatched entries in v's row, divided by $U_{t-1,b,b'} - i - |\overline{E}^{t-1}| \cdot (d/2)$. The term $U_{t-1,b,b'} - i$ is the number of unmatched entries after i steps, and the term $|\overline{E}^{t-1}| \cdot (d/2)$ is the maximum number of unmatched entries that cannot be matched to entries in u's row because the correspond to vertices w such that $(u, w) \in \overline{E}^{t-1}$.

Since $U_{t-1,b,b'} \geq (n/3) \cdot (d/2) - (t-1)$, $|\overline{E}^{t-1}| \leq t-1$ and $t - 1 < Q = o(n/d)$ the probability is $O(d/n)$. It follows that the probability that the algorithm gets a positive answer for *any* of the at most $Q = o(n/d)$ vertex-pair queries is $o(1)$. But if the algorithm always gets negative answers for its vertex-pair queries it clearly cannot close a triangle with such a query.

Neighbor queries. Given a neighbor query (u, b', i) where $u \in V_b$, the process gives an answer (v, j) where $v \in V_{b'}$ according to the conditional probability that the entries $T_{b,b'}[u, i]$ and $T_{b',b}[v][j]$ are matched (where again, the conditioning is on the current knowledge graph G^{t-1}). Here we would like to upper bound the probability that v already belongs to the knowledge graph. Observe that the knowledge graph G^{t-1} contains at most $2t$ vertices and at most t edges and non-edges. Hence the conditional probability that $T_{b,b'}[u, i]$ is matched to $T_{b',b}[v, j]$ for some $v \in V_{b'}^{t-1} = V^{t-1} \cap V_{b'}$ and $j \in \{1, \ldots, d/2\}$, is upper bounded by the following expression: $(d/2) \times (2t)$ (the maximum number of unmatched entries $T_{b',b}[v, j]$ for $v \in V_{b'}^{t-1}$) divided by the total number of unmatched entries in $T_{b',b}$ that do not belong to rows of vertices $u \in V_{b'}$ such that $(u, v) \in \overline{E}^{t-1}$, minus $d/2$

(where the argument is very similar to the one given for vertex-pair queries). By our assumption on Q (where $t-1 < Q$), the numerator in the above expression is $o(\min\{n, d^2\})$ and the denominator is at least $(n/3) \cdot (d/2) - (t-1) \cdot (d/2) - (d/2) = \Omega(n \cdot d)$. Hence, the probability that such an event occurs at a given particular time t is $o(\min\{(1/d), (d/n)\}$, and the probability that it occurs at any $t \leq Q$ is $o(1)$. But if the algorithm never gets a vertex in the knowledge graph as an answer to a neighbor query, it clearly cannot close a triangle, or any cycle, with such a query.

Finally we address the case that $d < \frac{2}{3\sqrt{3}}\sqrt{n}$. Recall that in this case we first partition the vertices into k parts $V^1, \ldots, V^k$ where $|V^i| = n' = \frac{27}{4}d^2$. We then apply the construction to each V^i. By the argument described above, here we can get a lower bound of the form $\Omega(\min\{d, n'/d\}) = \Omega(d)$. But for the current setting of d we also have that $\min\{d, n/d\} = d$, and so the lower bound holds in this case as well. ■ (Lemma 5 and Lemma 2)

A remark about multiple edges. The lower bound proof stated above is valid for graphs that may contain multiple edges. In the distribution D_Δ the probability of a multiple edge between a pair of vertices is $O(\frac{d^2}{n^2})$. Thus, graphs created according to this distribution contain multiple edges with probability close to 1. However, with probability $1 - o(1)$ there are $O(d^2)$ multiple edges in the graphs created according to this distribution. We have shown in Lemma 3 that graphs created according to the distribution D_Δ are $\Omega(1)$-far from being triangle-free with probability $1 - o(1)$. Hence we can deduce that by removing multiple edges from a graph G constructed according to the distribution D_Δ, the resulting graph is $\Omega(1)$-far from being triangle-free with probability $1 - o(1)$. In addition, an algorithm ALG that interacts with the process detects a multiple edge with probability $o(1)$ due to the following reason: ALG doesn't detect any edges by vertex-pair queries with probability $1 - o(1)$. The probability of detecting a multiple edge in a neighbor query is at most the probability that such a query is answered by a vertex in the knowledge graph. This probability was shown in Lemma 3 to be $o(1)$. Thus, at the end of the interaction with ALG, the process can delete all the multiple edges from the resulting graph, so that the resulting graph contains no multiple edges. The graph remains $\Omega(1)$-far from being triangle-free after the deletion, and its average degree is $d - o(1)$. As a conclusion we get that our lower bound is valid also for graphs with no multiple edges.

5 An Improved Lower Bound for High Degrees

In this section we establish the following lemma, which improves on our previous lower bound of $\min\{d, n/d\}$ when the degree of the graph is at least $n^{2/3+o(1)}$.

Lemma 6 *Every one-sided error testing algorithm for triangle-freeness must perform* $\Omega\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$ *queries, where* $\nu(n) = \frac{\log\log\log n + 4}{\log\log n}$. *This lower bound holds even for d-regular graphs.*

In order to prove the lemma, here too we define a distribution over graphs that are far from being triangle free. We then prove a lower bound on the number of queries that are required in order to detect a triangle with probability bounded away from zero in a graph that is generated

according to the distribution. As we shall see, it will actually be convenient to consider graphs over $3n$ vertices and degree $2d$.

5.1 A variant of Behrend graphs

Our lower bound distribution builds on graphs that are variants of what are known as *Behrend Graphs* [Beh46, RS76, SS42]. These graphs are defined by sets of integers that include no three-term arithmetic progression (abbreviated as 3AP). Namely, these are sets $X \subset \{1, \ldots, n\}$ such that for every three elements $x_1, x_2, x_3 \in X$, if $x_2 - x_1 = x_3 - x_2$ (i.e., $x_1 + x_3 = 2x_2$), then necessarily $x_1 = x_2 = x_3$. Below we describe a construction of such sets that are large (relative to n), and later explain how such sets determine Behrend graphs. Our construction of X uses similar ideas to those used in known constructions [Beh46, SS42] and gives a slightly weaker result. However, our alternative construction is somewhat simpler, and the size of the resulting set suffices for our purposes.

Lemma 7 *For every sufficiently large n there exists a set $X \subset \{1, \ldots, n\}$, $|X| \geq n^{1-\frac{\log\log\log n+4}{\log\log n}}$, such that X contains no three-term arithmetic progression.*

Proof: For simplicity we we do not explicitly write floors (or ceilings). Let $b = \log n$ and $k = \log n / \log b - 2$. Since $\log n / \log b = \log n / \log\log n$ we have that $k < b/2$ for every $n \geq 8$. We arbitrarily select a subset of k different numbers $\{x_1, \ldots, x_k\} \subset \{0, \ldots, b/2 - 1\}$ and define $X = \left\{\sum_{i=1}^{k} x_{\pi(i)} b^i \ : \ \pi \text{ is a permutation of } \{1, \ldots, k\}\right\}$. By the definition of X we have that $|X| = k!$. By using $z! > (z/e)^z$, we get that

$$|X| \ = \ k! \ \geq \ \left(\frac{\log\log n}{\log n}\right)^2 \cdot \left(\frac{\log n}{\log\log n}\right)! \ > \ n^{1-\frac{\log\log\log n+4}{\log\log n}} \tag{35}$$

Consider any three elements $u, v, w \in X$ such that $u + v = 2w$. By definition of X, these elements are of the form $u = \sum_{i=1}^{k} x_{\pi_u(i)} b^i$, $v = \sum_{i=1}^{k} x_{\pi_v(i)} b^i$ and $w = \sum_{i=1}^{k} x_{\pi_w(i)} b^i \in X$, where π_u, π_v, π_w are permutations over $\{1, \ldots, k\}$. Since $x_i < b/2$ for every $1 \leq i \leq k$, it must be the case that for each i,

$$x_{\pi_u(i)} + x_{\pi_v(i)} = 2x_{\pi_w(i)} \tag{36}$$

This implies that for every i:

$$x^2_{\pi_u(i)} + x^2_{\pi_v(i)} \geq 2x^2_{\pi_w(i)} \tag{37}$$

where the inequality in Equation (37) is strict unless $x_{\pi_u(i)} = x_{\pi_v(i)} = x_{\pi_w(i)}$. If we sum over all i's and there is at least one index i for which the inequality in Equation (37) is strict we get that

$$\sum_{i=1}^{k} x^2_{\pi_u(i)} + \sum_{i=1}^{k} x^2_{\pi_v(i)} > \sum_{i=1}^{k} 2x^2_{\pi_w(i)} \tag{38}$$

which is a contradiction since we took permutations of the same numbers. Thus, we get that $u = v = w$. ■

Remark. In fact, the set constructed above is also 3AP-free when all calculations are performed modulo n. We will use this observation below.

Behrend graphs. Given a set $X \subset \{1,\ldots,n\}$ with no three-term arithmetic progression we define the Behrend graph BG_X as follows. It has $3n$ vertices that are partitioned into three equal parts: V_1, V_2, and V_3. For each $i \in \{1,2,3\}$ we associate with each vertex in V_i a different integer in $\{0,\ldots,n-1\}$. The edges of the graph are defined as follows:

- The edges between V_1 and V_2: For every $x \in X$ and $j \in \{0,\ldots,n-1\}$ there is an edge between $j \in V_1$ and $(j+x) \bmod n \in V_2$;
- The edges between V_2 and V_3: For every $x \in X$ and $j \in \{0,\ldots,n-1\}$ there is an edge between $(j+x) \bmod n \in V_2$ and $(j+2x) \bmod n \in V_3$;
- The edges between V_1 and V_3: For every $x \in X$ and $j \in \{0,\ldots,n-1\}$ there is an edge between $j \in V_1$ and $(j+2x) \bmod n \in V_3$.

We shall say that an edge between $j \in V_1$ and $j' \in V_2$ or between $j \in V_2$ and $j' \in V_3$ is *labeled* by x, if $j' = (j+x) \bmod n$, and we shall say that an edge between $j \in V_1$ and $j' \in V_3$ is *labeled* by x, if $j' = (j+2x) \bmod n$.

The graph BG_X is $2|X|$-regular and it contains $3|X|n$ edges. For every $j \in \{0,\ldots,n-1\}$ and $x \in X$, the graph contains a triangle $(j, (j+x) \bmod n, (j+2x) \bmod n)$ where $j \in V_1$, $(j+x) \bmod n \in V_2$ and $(j+2x) \bmod n \in V_3$. There are $n \cdot |X|$ such edge-disjoint triangles and every edge is part of one such triangle. Moreover, there are no other triangles in the graph. To verify this consider any three vertices j_1, j_2, j_3 where $j_i \in V_i$ and such that there is a triangle between the three vertices. By definition of the graph, $j_2 = (j_1 + x_1) \bmod n$, for some $x_1 \in X$. $j_3 = (j_2 + x_2) \bmod n$, for some $x_2 \in X$. $j_3 = (j_1 + 2x_3) \bmod n$, for some $x_3 \in X$. Therefore, $(j_1 + x_1 + x_2) \bmod n = (j_1 + 2x_3) \bmod n$. That is, we get that $(x_1 + x_2) \bmod n = (2x_3) \bmod n$. Note that by the definition of X, for every $x \in X$, $x < n/2$, and so $x_1 + x_2 = 2x_3$. Since X contains no three-term arithmetic progression, the last implies that $x_1 = x_2 = x_3$, meaning that the triangle (j_1, j_2, j_3) is of the form $(j, (j+x) \bmod n, (j+2x) \bmod n)$.

5.2 The edge density of large sets in random Behrend graphs

In this subsection we prove the following lemma, which is central to the proof of Lemma 6. We shall use the following notation: For a subset $Y \subseteq X$ and a subset of vertices C in BG_Y, we let $e_Y(C)$ denote the number of edges spanned by C in BG_Y.

Lemma 8 *Let $0 < \beta < \frac{1}{2}$ and $0 < \alpha \leq 1$ be such that $\alpha - 2\beta > \frac{1}{\log\log n}$, and let $X \subset \{1,\ldots,n\}$, $|X| \geq n^\beta$. Consider the random Behrend graph BG_Y obtained by choosing a random subset $Y \subseteq X$, $|Y| = d = \frac{|X|}{n^\beta}$. With high probability over the choice of Y, for every subset C of vertices in BG_Y where $|C| = n^\alpha$, we have $e_Y(C) \leq \frac{90}{\alpha - 2\beta}\frac{n^{2\alpha}}{n^\beta}$ edges.*

The lemma states that for sufficiently large subsets C (i.e., for $|C| = n^\alpha$, where $\alpha - 2\beta$ is a constant), the number of edges $e_Y(C)$ is not much larger than its expected value. Note that the smaller we choose β (i.e., the larger we choose Y), the smaller can α be. Thus, the lemma can be applied to sets with of relatively small size.

Before proving the lemma we introduce some notation and prove two claims. For a subset $W \subseteq V_1 \cup V_2$, $|W| = s$, let $W_1 = W \cap V_1$, $W_2 = W \cap V_2$ and consider the subgraph of BG_X induced

by W. Let

$$\begin{aligned}\Delta(W) &= \{(j_2 - j_1) \bmod n \,:\, j_1 \in W_1,\ j_2 \in W_2, \\ &\qquad \text{and } (j_2 - j_1) \bmod n \in X\} \end{aligned} \tag{39}$$

denote the set of *differences* in W. That is, it is the set of labels of the edges between W_1 and W_2 in BG_X. Obviously, $|\Delta(W)| \leq |W|^2 = s^2$. For every difference $x \in \Delta(W)$, we define the *multiplicity* of x in W as the number of edges in BG_X between vertices in W_1 and vertices in W_2 that are labeled by x.

Let $k = \frac{5}{\alpha - 2\beta}$. For β and α that satisfy the condition of the lemma ($\alpha - 2\beta > \frac{1}{\log\log n}$) we have that $k \leq 5 \log\log n$. We shall say that W is *good* if no difference in $\Delta(W)$ has multiplicity higher than k in W.

Claim 9 *With high probability over the choice of $Y \subseteq X$, for every good W such that $|W| = s \geq n^\beta \log n$, we have that $e_Y(W) \leq \frac{2ks^2}{n^\beta}$.*

Proof: Consider a fixed choice of a good W such that $|W| = s = n^\beta \log n$. By definition, $|\Delta(W)| \leq |W|^2 = s^2$. Since W is good, for every $x \in \Delta(W)$, if $x \in Y$, then the number of edges labeled by x in the subgraph of BG_X induced by W is at most k.

Since $Y \subseteq X$, $|Y| = d = \frac{|X|}{n^\beta}$ is a random subset selected uniformly from a set of size $|X|$, the expected size of $\Delta(W) \cap Y$ is $|\Delta(W)| \cdot n^{-\beta} \leq s^2 \cdot n^{-\beta}$. Using known bounds of the tail of the Hypergeometric distribution (see, e.g., [JuR00, Page 29]), the probability that $|\Delta(W) \cap Y| > 2 \cdot s^2 \cdot n^{-\beta}$ is upper bounded by $\exp\left(-\frac{cs^2}{n^{-\beta}}\right)$ for some constant c. The claim follows by taking a union bound over all choices of W of size s. ■

We also need the following claim.

Claim 10 *Let C be a subset of $V_1 \cup V_2$, such that $|C| = n^\alpha$. Suppose we uniformly and independently select $W \subset C$, $|W| = n^\beta \log n$. Then the probability that W is* not good *is at most $\frac{1}{n^\beta}$.*

Proof: Note that by the definition of Behrend graphs, the edges between vertices in C that are labeled by a specific difference, form a matching. When we choose a random subset $W \subset C$, the probability that there exists a single difference of C (i.e. an element of $\Delta(C)$) that has multiplicity at least $k+1$ in W is bounded by

$$|C|^2 |C|^{k+1} \binom{|C| - (2k+2)}{|W| - (2k+2)} \binom{|C|}{|W|}^{-1}. \tag{40}$$

To verify this expression, note first that there are at most $|C|^2$ possibilities to choose a difference from $\Delta(C)$. Since the edges of a specific difference form a matching of size at most $|C|$ over the edges of C, there are at most $|C|^{k+1}$ possibilities to choose the $k+1$ edges of this difference. The $k+1$ edges of the difference determine $2k+2$ of the vertices of W. Thus there are $\binom{|C|-(2k+2)}{|W|-(2k+2)}$ possibilities to choose the remaining vertices of W. Now,

$$\frac{|C|^2 |C|^{k+1} \binom{|C|-(2k+2)}{|W|-(2k+2)}}{\binom{|C|}{|W|}} \leq |C|^2 \cdot |C|^{k+1} \cdot \left(\frac{|W|}{|C|}\right)^{2k+2}$$

$$\leq \quad n^{2\alpha}\left(\frac{\log^2 n}{n^{\alpha-2\beta}}\right)^{\frac{5}{\alpha-2\beta}} \quad \leq \quad \frac{1}{n^2} \tag{41}$$

The last expression is upper bounded by $\frac{1}{n^\beta}$ as required. ■

Proof of Lemma 8. Consider a set C of vertices of BG_Y such that $|C| = n^\alpha$. Let $C_i = C \cap V_i$ for $1 \leq i \leq 3$. We will show that almost surely the number of edges between C_1 and C_2 is at most $\frac{30}{\alpha-2\beta}\frac{n^{2\alpha}}{n^\beta}$. The argument for the number of edges between C_2 and C_3 and between C_1 and C_3 is analogous, and hence the lemma follows. We shall prove the claim for every C_1, C_2 such that $|C_1 \cup C_2| = n^\alpha$. Clearly this implies that it holds for every C_1, C_2, s.t. $|C_1 \cup C_2| \leq n^\alpha$.

By Claim 9, with high probability over the choice of Y the following holds. For every good W, $W \subset C$, such that $|W| = s \geq n^\beta \log n$, the number of edges spanned by W in BG_Y is at most $2ks^2n^{-\beta}$. Assume from now on that the selected Y has this property. We shall use Claim 10 to derive an upper bound on the number of edges in BG_Y that are spanned by the vertices of C.

By our assumption on Y, if W is good and $|W| = s \geq n^\beta \log n$ then $e_Y(W) \leq 2ks^2n^{-\beta}$. Clearly, $e_Y(W) \leq s^2$. If we uniformly at random select $W \subset C$, such that $|W| = s$ then

$$\mathrm{Exp}[e_Y(W)] \geq \frac{1}{2} \cdot e_Y(C) \cdot \frac{s^2}{n^{2\alpha}} \, .$$

We stress that the expectation is taken only over the choice of W and not over the choice of Y. Now,

$$\begin{aligned}
&\mathrm{Exp}[e_Y(W)] \\
&= \mathrm{Exp}[e_Y(W) \mid W \text{ is good }] \cdot \Pr[W \text{ is good }] + \mathrm{Exp}[e_Y(W) \mid W \text{ is not good }] \cdot \Pr[W \text{ is not good }] \\
&\leq 2ks^2 \cdot n^{-\beta} + s^2 \cdot n^{-\beta} \;=\; (2k+1)s^2 \cdot n^{-\beta}
\end{aligned} \tag{42}$$

It follows that

$$e_Y(C) \leq (2k+1) \cdot 2|C|^2 \cdot n^{-\beta} \leq 5k|C|^2 \cdot n^{-\beta} \, . \tag{43}$$

Since $k = \frac{5}{\alpha-2\beta}$, the lemma follows. ■

As a corollary of Lemma 8 we get:

Corollary 11 *Let $0 < \beta < \frac{1}{2}$ and $X \subset \{1, \ldots, n\}$ where $|X| \geq n^{1-\nu(n)}$ for $\nu(n) = \frac{\log\log\log n + 4}{\log\log n}$. Consider the random Behrend graph BG_Y obtained by choosing a random subset $Y \subseteq X$, $|Y| = d = \frac{|X|}{n^\beta}$. With high probability over the choice of Y, for every subset C of vertices in BG_Y such that $|C| \leq \min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}$, the following bound applies: $|C| \cdot e_Y(C) \leq n^{1-\nu(n)}$.*

Proof: Note that $d = n^{1-\nu(n)-\beta}$, and so we need to bound the number of edges in sets C s.t.

$$|C| \leq \min\left\{n^{\frac{1-\nu(n)-\beta}{2}}, n^{\frac{1+\nu(n)+\beta}{3}}\right\} \cdot n^{-\nu(n)} \leq n^{2/5-\nu(n)}, \tag{44}$$

where $n^{2/5-\nu(n)}$ is obtained for $\beta = 1/5 - \nu(n)$.

Consider first the case that $\beta \leq \frac{1}{5} - \nu(n)$. In this case we need to bound the number of edges in sets C, s.t. $|C| \leq \frac{n^{2/3-\nu(n)}}{d^{1/3}} = n^{\frac{1-2\nu(n)+\beta}{3}}$.

Let $\alpha = \frac{1-2\nu(n)+\beta}{3}$ and observe that $\alpha - 2\beta \geq \nu(n) > \frac{1}{\log\log n}$. Hence we can apply Lemma 8 and get that with high probability over the choice of Y, for every subset C such that $|C| = n^{\alpha}$, we have that $e_Y(C) \leq \frac{90}{\alpha-2\beta} \cdot \frac{n^{2\alpha}}{n^{\beta}}$. Clearly this upper bound holds also for every subset C such that $|C| \leq n^{\alpha}$. Hence,

$$|C| \cdot e_Y(C) \leq \frac{90}{\alpha - 2\beta} \cdot \frac{n^{3\alpha}}{n^{\beta}} \leq \frac{90}{\nu(n)} \cdot n^{1-2\nu(n)} \leq n^{1-\nu(n)} \tag{45}$$

Consider now the case that $\beta > \frac{1}{5} - \nu(n)$. Note that we need to bound the number of edges in sets C such that $|C| \leq n^{2/5-\nu(n)}$. We have shown that the bound applies for the case that $\beta = \frac{1}{5} - \nu(n)$, and $|C| = n^{2/5-\nu(n)}$. Hence, for $\beta > \frac{1}{5} - \nu(n)$, the sets C for which $|C| \leq n^{2/5-\nu(n)}$ contain less edges and therefore the previous bound applies. ■

5.3 The lower bound distribution $BG(n, d)$

Let $X \subset [n]$ be a set with no three-term arithmetic progression, as constructed in Subsection 5.1, such that $|X| = n^{1-\nu(n)}$ (where $\nu(n) = \frac{\log\log\log n+4}{\log\log n}$). Consider the Behrend graph, denoted BG_X, whose set of generators is X. Recall that BG_X, which is a graph over $3n$ vertices, contains $|X| \cdot n$ edge-disjoint triangles: every edge belongs to exactly one triangle, and every triangle corresponds to some $x \in X$.

For each subset $Y \subset X$, such that $|Y| = d$ we consider the subgraph of BG_X that contains all its vertices but only the edges labeled by differences $y \in Y$. This (sub-)graph contains $n \cdot |Y| = nd$ edge-disjoint triangles and is hence ϵ-far from being triangle free for any $0 < \epsilon < 1/3$. Next we apply a permutation π on the names of the vertices. More precisely, π consists of 3 permutations, $\pi^{(b)}$, $b \in \{1, 2, 3\}$, each over $\{0, \ldots, n-1\}$. If we denote each vertex v in BG_X by a pair (b, i) where $b \in \{1, 2, 3\}$ is the index of the subset that the vertex belongs to and $i \in \{0, \ldots, n-1\}$, then $\pi(v) = \pi(b, i) = (b, \pi^{(b)}(i))$. We denote the resulting graph by $BG_{Y,\pi}$.

A graph is generated according to the distribution $BG(n, d)$ by uniformly selecting Y and π and outputting the resulting graph $BG_{Y,\pi}$. We also assume that the edges incident to a vertex v are ordered randomly in the incidence list of v. For the sake of simplicity, we do not include these random labelings in the notation.

5.4 Online generation of graphs according to $BG(n, d)$

In order to prove Lemma 6 we would like to show that any algorithm that is given query access to a graph generated according to $BG(n, d)$ must perform $\Omega\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$ queries in order to detect a triangle with sufficiently high constant probability. We wish to stress that the algorithm can be much more powerful/sophisticated potentially than just an algorithm that samples a random set of the input and looking for a triangle inside.

For the sake of our analysis, it will be convenient to define an online process, denoted P, that answers the algorithm's queries while generating a graph according to $BG(n, d)$. The process P will actually provide the algorithm with more information than required by neighbor and vertex-pair queries. Namely, whenever the algorithm asks a query involving a vertex $v \in \{(1, 0), \ldots, (3, n-1)\}$

(in either type of query), the process will provide it with $\pi^{-1}(v)$. This will also be the case when the process answers a neighbor query (u, i) with a vertex v. Thus the algorithm is provided with the "identity" in BG_X of the vertices it has observed, and can also derive the labels $y \in Y$ of the edges it has observed.

Clearly, a lower bound on the number of queries of an algorithm that is provided with the additional information described above constitutes a lower bound on an algorithm that is not provided with this additional information. It will actually be convenient to consider the following three types of queries:

1. Vertex queries: for any choice of a vertex $v \in \{(1,0), \ldots, (3, n-1)\}$ the algorithm is provided with $\pi^{-1}(v)$.
2. Random neighbor queries: for any vertex v the algorithm has already observed, it may ask for a new random neighbor u of v (together with $\pi^{-1}(u)$). The algorithm can indicate the subset to which the neighbor belongs.
3. Difference queries: for any $x \in X$ the algorithm can ask whether $x \in Y$.

Vertex queries are denoted $(VQ, (b, i))$, neighbor queries are denoted $(NQ, (b, i), b')$, and difference queries are denoted (DQ, x) where $b \neq b' \in \{1, 2, 3\}$, $i \in \{0, \ldots, n-1\}$, and $x \in X$.

Note that a vertex-pair query can be performed by asking at most two vertex queries and one difference query, and a neighbor query can be performed by asking a vertex query and a random neighbor query (recall that the edges adjacent to a vertex are labeled randomly in $BG_{Y,\pi}$). It follows that any lower bound on algorithms that perform these types of queries implies a lower bound that is at most a factor of 3 smaller on algorithms that perform vertex-pair and neighbor queries.

Let ALG be an algorithm that performs $Q = Q(n, d)$ queries of the above three types and whose goal is to detect a triangle with probability at least 9/10. The probability is taken over the choice of the graph $BG_{Y,\pi}$ and the coin flips of the algorithm. Namely, ALG is a (possibly probabilistic) mapping from *query-answer histories* $\langle (q_1, t_1), \ldots, (q_t, a_t) \rangle$, to q_{t+1} for every $t < Q$. The mapping needs to be defined only on histories that are consistent with some graph $BG_{Y,\pi}$.

As described above, a vertex query $q_t = (VQ, v_t)$ for $v_t \in \{(1,0), \ldots, (3, n-1)\}$ that has not yet been observed is answered by $\pi^{-1}(v_t)$. A random neighbor query $q_t = (NQ, v_t, b')$ for $v_t = (b, i)$ that has been observed is answered by a new random neighbor $U_t = (b', j)$ of v_t together with $\pi^{-1}(U_t)$. A difference query $q_t = (DQ, x)$ for $x \in X$ is answered by '1' (yes) or '0' (no), indicating whether $x \in Y$ or not.

Any query-answer history of length t can be used to define the *knowledge base* $K_t = (V_t, Y_t, \overline{Y}_t, \pi_t)$, where $V_t \subset \{(1,0), \ldots, (3, n-1)\}$, $Y_t, \overline{Y}_t \subset X$ and $\pi_t : V_t \to \{(1,0), \ldots, (3, n-1)\}$ (where π_t is one-to-one). Specifically, V_t consists of all vertices (b, j) such that $(b, j) = \pi^{-1}(v)$ for some v that appeared either in one of the first t queries of ALG or in one of the first t answers. The set Y_t consists of all $x \in X$ such that for some $t' \leq t$ there was either a query $q_{t'} = (DQ, x)$ that was answered by '1', or a query $q_{t'} = (NQ, v_t, b')$ that was answered with U_t where the edge between $\pi^{-1}(v_t)$ and $\pi^{-1}(U_t)$ is labeled by x. The set $\overline{Y}_t$ consists of all $x \in X$ such that for some $t' \leq t$ there was a query $q_{t'} = (DQ, x)$ that was answered by '0'. Finally, for every $(b, j) \in V_t$, $\pi_t(b, j) = v$ where v is such that $\pi^{-1}(v) = (b, j)$.

Observe that V_t together with Y_t determine a subgraph of BG_X: the vertices of the subgraph are the vertices of V_t, and the edges are all pairs (u,v), $u,v \in V_t$ such that there is an edge between u and v in BG_X and this edge is labeled by some $x \in Y_t$. For each $b \in \{1,2,3\}$ we let $V_{t,b} = \{(b,j) : (b,j) \in V_t\}$.

5.4.1 Definition of the process P.

Let $R = R(n,d)$ denote the set of all graphs $BG_{Y,\pi}$ in the support of $BG(n,d)$. For $b \in \{1,2,3\}$ and $i,j \in \{0,\ldots,n-1\}$ let $R_{b,i,j} \subset R$ denote the subset of graphs $BG_{Y,\pi} \in R$ such that $\pi(b,j) = (b,i)$. For $x \in X$ let $R_x \subset R$ denote the subset of graphs $BG_{Y,\pi} \in R$ such that $x \in Y$, and let $R_{\neg x} \subset R$ denote the subset of graphs $BG_{Y,\pi} \in R$ such that $x \notin Y$.

The process P answers ALG's queries as follows, where we assume without loss of generality that ALG does not ask any query q_t whose answer can be deduced from the knowledge base K_{t-1}. In particular, for every vertex query $q_t = (VQ, v)$ we have that $v \notin V_{t-1}$, and for every difference query $q_t = (DQ, x)$ we have that $x \notin Y_{t-1} \cup \overline{Y}_{t-1}$. The process P initializes $R_0 = R$, and in general, for any $t \geq 0$, we have that R_t consists of all graphs in R that are consistent with the first t queries.

- To answer a vertex query $q_t = (VQ, v)$, where $v = (b,i)$, the process uniformly selects $(b,j) \in \{(b,0),\ldots,(b,n-1)\} \setminus V_{t-1,b}$ and sets $R_t = R_{b,i,j} \cap R_{t-1}$. Note that (b,j) is selected with probability $\frac{|R_{b,i,j} \cap R_{t-1}|}{|R_{t-1}|} = \frac{|R_t|}{|R_{t-1}|}$.
- Given a difference query (DQ, x), with probability $\frac{d-|Y_{t-1}|}{|X|-|Y_{t-1}|-|\overline{Y}_{t-1}|}$ the process answers '1' and with probability $1 - \frac{d-|Y_{t-1}|}{|X|-|Y_{t-1}|-|\overline{Y}_{t-1}|} = \frac{|X|-d-|\overline{Y}_{t-1}|}{|X|-|Y_{t-1}|-|\overline{Y}_{t-1}|}$ it answers '0'. In the former case it sets $R_t = R_x \cap R_{t-1}$ and in the latter case it sets $R_t = R_{\neg x} \cap R_{t-1}$. Note that here the answer is '1' with probability $\frac{|R_x \cap R_{t-1}|}{|R_{t-1}|} = \frac{|R_t|}{|R_{t-1}|}$ and is '0' with probability $1 - \frac{|R_x \cap R_{t-1}|}{|R_{t-1}|} = \frac{|R_{\neg x} \cap R_{t-1}|}{|R_{t-1}|} \frac{|R_t|}{|R_{t-1}|}$.
- To answer a random neighbor query $q_t = (NQ, v, b')$, the process performs two steps. First it selects $x \in X$ in the following manner. With probability $\frac{|Y_{t-1}|}{d}$ it selects x uniformly from Y_{t-1}, and with probability $1 - \frac{|Y_{t-1}|}{d}$ it selects x uniformly from $X \setminus (Y_{t-1} \cup \overline{Y}_{t-1})$. In either case the choice of x (together with b') determines the value of $j \in \{0,\ldots,n-1\}$ such that there is an edge labeled x in BG_X between $\pi^{-1}(v)$ and (b',j). If $(b',j) \in V_{t-1}$ then $u = \pi_t(b',j) = \pi(b',j)$ is known and $R_t = R_x \cap R_{t-1}$. Otherwise $u = (b',i)$ is selected uniformly in $\{(b',0),\ldots,(3,n-1)\} \setminus V_{t-1,b'}$ and $R_t = R_x \cap R_{b',i,j} \cap R_{t-1}$.

Once P completes answering the Q queries of ALG it uniformly selects a graph in R_Q. The next lemma is easily derived from the definition of the process.

Lemma 12 *For every algorithm ALG, the process P, when interacting with ALG, uniformly generates graphs in $BG(n,d)$*

Proof: Consider a specific graph G in $R_0 = BG(n,d)$. The probability that G is generated by P is

$$\Pr[G \in R_1] \cdot \Pr[G \in R_2 | G \in R_1] \cdots \Pr[G \in R_Q | G \in R_{Q-1}] \cdot \frac{1}{|R_Q|}$$

$$= \frac{|R_1|}{|R_0|} \cdot \frac{|R_2|}{|R_1|} \cdot \cdots \cdot \frac{|R_Q|}{|R_{Q-1}|} \cdot \frac{1}{|R_Q|} = \frac{1}{|R_0|} \tag{46}$$

and the lemma follows. ■

5.5 Proof of Lemma 6

Consider any algorithm ALG that interacts with the process P. We shall show that if ALG asks $Q = o\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$ queries, then the probability that it reveals a triangle is $o(1)$. By Corollary 11, with high probability over the choice of Y, $|Y| = d$, for every subset U of vertices such that $|U| = o\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$, we have $|U| \cdot e(U) = o(n^{1-\nu(n)})$, where $e(U)$ denotes the number of edges induced by the vertices of U. In what follows we assume that the graph generated by P in fact has this property, where we take into account the probability of $o(1)$ that this is not the case. Let E_t be the set of edges between vertices in V_t that are known to the algorithm after the first t queries. That is, E_t consists of all edges in BG_X whose labels are in Y_t and are between vertices in V_t. Therefore, for every $t = o\left(\min\left\{\sqrt{d}, \frac{n^{2/3}}{d^{1/3}}\right\} \cdot n^{-\nu(n)}\right)$ we have that $t|E_t| = o(n^{1-\nu(n)})$.

Recall that every edge $(i,j) \in E_t$ participates in exactly one triangle. There are two ways by which ALG can close a triangle in its t-th query. If the query is either a vertex query or a random neighbor query, the algorithm must receive as an answer one of the $|E_{t-1}|$ vertices that close triangles with (the known) edges between vertices in V_{t-1}. If the query is a difference query (DQ, x) (where $x \notin Y_{t-1} \cup \overline{Y}_{t-1}$), then there must be three vertices $(1,i),(2,j),(3,k) \in V_{t-1}$ that form a triangle in BG_X whose edges are labeled by x and the answer to the query is '1' (i.e., $x \in Y$). For sake of simplicity we assume that whenever the algorithm obtains a vertex that closes a triangle in BG_X, then it is also told whether this triangle is in $BG_{Y,\pi}$ or not (i.e., it gets a difference query "for free"). We now turn to bounding the probability of each of the above events by which a triangle is closed.

We first observe that since $|V_t| \leq t = o(n)$, whenever ALG asks a vertex query (VQ, v) where $v = (b,i)$, the answer, (b,j), is uniformly distributed in a subset of size $\Theta(n)$. Since $|Y_t| + |\overline{Y}_t| \leq t$, whenever ALG asks a random neighbor query $q_t = (NQ, v, b')$, with probability at least $1 - \frac{t}{d}$ we have that $(b', j) = \pi^{-1}(u)$ is uniformly distributed in a subset of size $|X| - t$. Since $|X| = \Omega(n^{1-\nu(n)})$, and $t = o(\sqrt{d}) = o(\sqrt{|X|})$, with probability $1 - o(1)$, for every neighbor query performed, the answer to the neighbor query is uniformly distributed in a subset of size $\Omega(n^{1-\nu(n)})$. Let us assume from this point on that this is in fact the case (where we take into account that $o(1)$ probability that this is not the case).

Given the above, the probability that ALG closes a triangle in the t-th query when one of the edges of the triangle is already in E_t is $\frac{|E_t|}{\Omega(n^{1-\nu(n))}}$. It remains to bound the probability that ALG closes a triangle by obtaining a vertex (b,j) that closes a triangle in the subgraph of BG_X that is induced by V_{t-1}, and that the difference $x \notin Y_{t-1} \cup \overline{Y}_{t-1}$ corresponding to the triangle is determined to be in Y. Since the number of edges in the subgraph of BG_X that is induced by V_{t-1} is upper bounded by t^2, the probability of the above event is at most

$$\frac{t^2}{\Omega(n^{1-\nu(n))})} \cdot \frac{d-t}{\Omega(n^{1-\nu(n))})} = \frac{t^2(d-t)}{\Omega(n^{2(1-\nu(n))})} \,. \tag{47}$$

Since $t = o\left(\min\{\sqrt{d}, n^{2/3}/d^{1/3}\} \cdot n^{-\nu(n)}\right)$, we know that $t|E_t| = o(n^{1-\nu(n)})$ and $t^3 \cdot (d-t) = o(n^{2(1-\nu(n))})$. Hence, the probability that one of the above events occurs for any $t \leq Q$ is $o(1)$, as required. ■

6 From 1-Sided Error to 2-Sided Error

In this section we establish that under certain conditions, a one-sided error lower bound for triangle-freeness can be transformed into a two-sided error lower bound. Since these conditions hold for our one-sided error lower bounds, we obtain two-sided error lower bounds.

Theorem 3 *Let D_Δ be a distribution over graphs with n vertices and average degree d, and let $q(n,d)$ be a function of these parameters. Assume the following holds:*

- *With probability $1-o(1)$ a graph selected according to D_Δ is ϵ-far from being triangle-free for some constant ϵ.*
- *One of the following two conditions holds:*
  1. *In all graphs in the support of D_Δ, the triangles are edge-disjoint, and for any algorithm A, the probability that A reveals a triangle in a graph selected according to D_Δ using $o(q(n,d))$ queries is less than $2/3$.*
  2. *For any algorithm A, the probability that A reveals a cycle (of any length) in a graph selected according to D_Δ using $o(q(n,d))$ queries is less than $2/3$.*

Then any two-sided error algorithm for testing triangle-freeness that has success probability at least $5/6$ must perform $\Omega(q(n/2,d))$ queries.

Since the distributions that are defined for our one-sided error lower bounds, which are stated in Lemmas 2 and 6, are as required by Theorem 3, we get the following corollary.

Corollary 13 *Any algorithm for testing triangle-freeness must perform*

$$\Omega\left(\max\left\{\min\{d, n/d\}, \min\left\{\sqrt{d}, n^{2/3}/d^{1/3}\right\} \cdot n^{-\nu(n)}\right\}\right)$$

queries. This lower bound holds even if the algorithm is allowed two-sided error and $d_{\max} = O(d)$.

6.1 Proof of Theorem 3: Condition 1

Given the distribution D_Δ we define two distributions over graphs that have $n' = 2n$ vertices and average degree d. One distribution, denoted D'_Δ, generates graphs that are ϵ-far from being triangle-free, and the other distribution, denoted $D_{\bar{\Delta}}$ generates graphs that are triangle-free. Assume, contrary to what is claimed in the theorem that there exists a two-sided error algorithm A' for testing triangle freeness that performs $o(q(n'/2,d))$ queries and has success probability at least $5/6$. Then, in particular, using $o(q(n'/2,d)) = o(q(n/d))$ queries, A' should be able to distinguish

with sufficiently high probability between graphs generated by D'_{Δ} and graphs generated by $D_{\bar{\Delta}}$. We shall show that we can then use A' to obtain an algorithm A that performs $o(q(n,d))$ queries and with probability at least $2/3$ reveals a triangle in a random graph generated according to D_{Δ}.

Defining the two distributions. In both distributions, a graph G' over $n' = 2n$ vertices is generated by first selecting a graph G from D_{Δ}. Every vertex v in G is replaced by two vertices, v_0 and v_1. Every edge (u, v) in G is replaced by two edges: either the two edges (u_0, v_0) and (u_1, v_1) (so that they are "in parallel") or the two edges (u_0, v_1) and (u_1, v_0) (so that they are "crossing"). If v is the j-th neighbor of u and u is the ℓ-th neighbor of v, then in both cases we maintain the ordering on neighbors. Namely, in the case of parallel edges we have that v_0 is the j-th neighbor of u_0 and v_1 is the j-th neighbor of u_1, u_0 is the ℓ-th neighbor of v_0 and u_1 is the ℓ-th neighbor of v_1 (an analogous correspondence holds for crossing edges). The difference between the distributions is in the choice (distribution on the choice) between the above two options.

Recall that the triangles in G are edge-disjoint. Hence, for each triangle in G, the edges between the corresponding vertices in G' can be determined independently from the edges that belong to other triangles. Consider a particular triangle (u, v, w) in G. There are $2^3 = 8$ ways to select the edges between the vertices $u_0, u_1, v_0, v_1, w_0, w_1$ (depending on whether we select parallel or crossing edges). In 4 of these ways we get 2 edge-disjoint triangles (e.g., (u_0, v_0, w_0) and (u_1, v_1, w_1)), and in 4 of these ways we get a single cycle of length 6 (e.g. $(u_0, v_0, w_0, u_1, v_1, w_1)$). The graph generated by D'_{Δ} simply selects one of the former 4 ways uniformly, and the graph generated by $D_{\bar{\Delta}}$ selects one of the latter 4 ways uniformly. For an illustration see Figure 2.

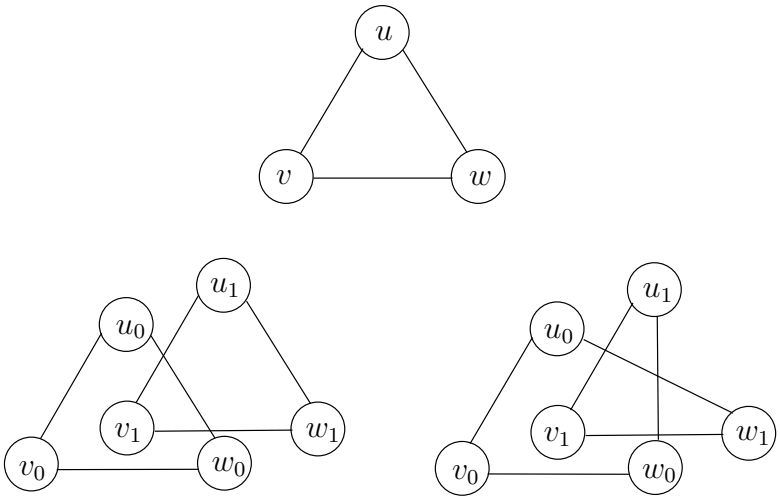

Figure 2: On the top is a triangle (u, v, w). On the bottom left is a transformation of this triangle into two disjoint triangles, and on the bottom right is a transformation into a cycle of length 6. In particular, on the left all pairs of edges are parallel, and on the right two are parallel and one is crossing.

The basic, but important observation is that for both distributions the following holds: If we consider any edge that belongs to a particular triangle in G, then the probability that the corresponding pair of edges in G' are parallel is equal to the probability that they are crossing. Moreover, this remains true if we condition on any other (single) edge in the triangle being transformed to either parallel or crossing edges. Independence breaks down only when we consider all 3 edges in

a triangle. We shall refer to this observation as the *Independence Observation*.

Using a two-sided error algorithm to find triangles. Let A' be a two-sided error algorithm for testing triangle freeness that performs $o(q(n'/2, d))$ queries when testing graphs over $n' = 2n$ vertices and has success probability at least $5/6$. We next show how to use it in order to detect triangles in a graph G over n vertices that is generated randomly according to D_Δ. The idea is that by performing queries to G and flipping some coins, we shall actually be emulating the execution of A' on graphs generated by either D'_Δ or $D_{\bar{\Delta}}$. Since A' is supposed to test graphs over $n' = 2n$ vertices, we denote the vertices in the queries it performs by $\{v_{1,0}, v_{1,1}, \ldots, v_{n,0}, v_{n,1}\}$.

Let G be a graph generated according to D_Δ. Algorithm A (whose goal is to detect a triangle in G) runs A' as a subroutine and answers its queries by performing queries to G and transforming the answers to the queries in an appropriate manner described below. In this process A maintains a *knowledge graph*, denoted $\widehat{G}$, which contains all the edges it has observed in G as well as the "non-edges" (i.e., pairs (u, v) that do not have an edge between them). In addition, A records all the answers it has already given to A'.

Whenever A' performs a degree query for a vertex $v_{i,b}$ ($b \in \{0, 1\}$), algorithm A queries the degree of v_i and returns it as an answer. Whenever A' performs a vertex-pair query $(v_{i,b}, v_{j,b'})$ $(b, b' \in \{0, 1\})$, if (u, v) is an edge or a non-edge in the knowledge graph $\widehat{G}$ then the answer to A' is determined. If this is not the case then A performs the vertex-pair query (v_i, v_j). If the answer is that there is no edge between the two vertices, then the answer given to A' is "no" as well. If the answer is that there is an edge, then there are two cases. If this edge closes a triangle with two other edges in $\widehat{G}$ then A terminates successfully. Otherwise, with probability $1/2$ A answers that there is an edge between $v_{i,b}$ and $v_{j,b'}$ and with probability $1/2$ it answers that there is no such edge. In both cases the existence of the edge in G is recorded in the knowledge graph $\widehat{G}$. In addition, in the former case A' is provided with the information concerning which neighbor is $v_{j,b'}$ of $v_{i,b}$.

Whenever A' performs a neighbor query $(v_{i,b}, \ell)$ (that does not correspond to an edge already in $\widehat{G}$), algorithm A performs the neighbor query (v_i, ℓ). Let the answer be (v_j, t). Namely, there is an edge between v_i and v_j, where v_j is the ℓ-th neighbor of v_i, and v_i is the t-th neighbor of v_j. Here too, if a triangle in G is detected then A terminates successfully. Otherwise it answers the query of A' in an analogous manner to the way a vertex-pair query is answered. If A' terminates before A has found a triangle, then A terminates unsuccessfully.

Completing the proof. Since A always terminates when or before it finds a triangle, by the Independence Observation, the distribution on the answers it gives to the queries of A' is exactly the same as the one we would get if the queries of A' were answered by a graph that is selected either according to D'_Δ or according to $D_{\bar{\Delta}}$. We claim that this implies that the probability that A' terminates before A finds a triangle (thus causing A to terminate unsuccessfully) is less than $1/3$. Here the probability is taken over the choice of G, the coin flips of A and the possible coin flips of A'.

Assume, contrary to the claim, that the probability that A' terminates before A finds a triangle is at least $1/3$. Consider the distribution over graphs that results from selecting with probability $1/2$ a graph G' according to D'_Δ, and with probability $1/2$ a graph G' according to $D_{\bar{\Delta}}$. By our counter assumption the probability that A' terminates before it sees three edges of the form (v_{i,b_1}, v_{j,b_2}),

(v_{j,b_3}, v_{k,b_4}) and (v_{k,b_5}, v_{i,b_6}) (where $b_1, \ldots, b_6 \in \{0,1\}$) is greater than $1/3$. In such a case, by the Independence Observation, the distribution on the answers to the queries of A' (and hence on its queries conditioned on these answers) is the same if the graph G' is selected according to D'_Δ or according to $D_{\bar{\Delta}}$. Therefore, the probability that A' terminates with an incorrect output, is greater than $1/6 - o(1) > 1/9$, where the term $o(1)$ is due to the probability that G is not ϵ-far from triangle-free. But this contradicts our assumption on A'.

Since the number of queries performed by A before it terminates is upper bounded by the number of queries performed by A', the theorem follows. ■

6.2 Proof of Theorem 3: Condition 2

Similarly to the proof of Theorem 3 under Condition 1, given the distribution D_Δ we define two distributions over graphs that have $n' = 2n$ vertices and average degree d. One distribution, denoted D'_Δ, generates graphs that are $\Omega(1)$-far from being triangle-free, and the other distribution, denoted $D_{\bar{\Delta}}$ generates graphs that are triangle-free. In fact, the graphs in the support of $D_{\bar{\Delta}}$ are all bipartite. We then show how it is possible to use an algorithm A' that can distinguish with high probability between graphs generated according to D'_Δ and graphs generated according to $D_{\bar{\Delta}}$ in order to obtain an algorithm A that with high probability reveals a cycle in a random graph generated according to D_Δ.

Defining the two distributions. In both distributions, a graph G' over $n' = 2n$ vertices is generated by first selecting a graph G from D_Δ. Here too, each edge $(u,v) \in E(G)$ is replaced by two edges in G': either the "parallel" edges (u_0, v_0) and (u_1, v_1) or the "crossing" edges (u_0, v_1) and (u_1, v_0). The difference between the distributions is in the choice (distribution on the choice) between the above two options.

In the distribution D'_Δ, the decision whether an edge is transformed into parallel edges or into crossing edges is done independently, and with equal probability. Namely, for each function $\sigma : E(G) \to \{p, c\}$, there is a graph G_σ in the support of D'_Δ. For every edge $(u,v) \in E(G)$, if $\sigma(u,v) = p$, then in G_σ we have the parallel edges (u_0, v_0) and (u_1, v_1), and if $\sigma(u,v) = c$, then in G_σ we have the crossing edges (u_0, v_1) and (u_1, v_0). The function σ is selected uniformly at random.

On the other hand, each graph in the support of $D_{\bar{\Delta}}$ is determined by a function $\pi : V(G) \to \{\ell, r\}$. Here ℓ stands for 'left' and r for 'right' so that π defines a two-way partition of the vertices: If $\pi(v) = \ell$ then v_0 is put on the 'left' side, and if $\pi(v) = r$ then v_0 is put on the 'right' side. In either case, v_1 is put on the opposite side. The decision between parallel and crossing edges is done so that the resulting graph is bipartite. Specifically, for every edge $(u,v) \in E(G)$, if $\pi(u) = \pi(v)$ then in G_π we have the crossing edges (u_0, v_1) and (u_1, v_0), and if $\pi(u) \neq \pi(v)$ then we have the parallel edges (u_0, v_0) and (u_1, v_1). The function π is selected uniformly at random. We note that different choices of π may result in the same graph, but this is inconsequential to our argument.

Properties of the distributions. By definition of $D_{\bar{\Delta}}$, every graph in the support of $D_{\bar{\Delta}}$ is bipartite, and hence is necessarily triangle-free. On the other hand, we claim that if the graph G (that is selected according to D_Δ) is ϵ-far from being triangle-free, then with probability $1 - \exp(-\Omega(\epsilon dn))$ over the choice of σ, the graph G_σ (in the support of D'_Δ) is at least $\epsilon/4$-far from being triangle-free. To verify this claim observe that since G is ϵ-far from being triangle-free, it contains at least ϵdn edge-disjoint triangles. For each such triangle, the probability that it is transformed into two

triangles in in G_σ is $1/2$. By a multiplicative Chernoff bound, the probability that less than $(\epsilon dn)/4$ of the disjoint triangles in G are transformed into a pair of disjoint triangles is $\exp(-\Omega(\epsilon dn))$. That is, with probability $1-\exp(-\Omega(\epsilon dn))$ there are $(\epsilon dn)/2$ disjoint triangles in the resulting graph, and since the graph contains $2n$ vertices, it is $(\epsilon/4)$-far from being triangle-free. Since the probability that G is ϵ-far from triangle free for some constant ϵ is $1-o(1)$, we get that with probability $1-o(1)$, a graph generated according to D'_Δ is at least $\Omega(1)$-far from being triangle-free.

Completing the proof. Most parts of the argument are almost identical to those in the proof of Theorem 3 subject to Condition 1, where the only difference is that "detecting a triangle" is replaced by "detecting a cycle". The only essential difference is that the "Independence Observation" concerning the distribution on parallel/crossing edges conditioned on a triangle not being detected, needs to be modified. Specifically, consider any fixed choice of edges in G that do not contain a cycle. We shall show that under both distributions, all mappings of these edges to parallel/crossing pairs of edges in G' have equal probability. This implies that if we fix any choice of parallel/crossing pairs of edges in G', where the corresponding edges in G do not contain a cycle, and we consider a new edge that does not close a cycle with these edges, then under both distributions it has equal conditional probability to be mapped to a parallel/crossing pair of edges in G'.

Let $F = \{(u_1, v_1), \ldots, (u_t, v_t)\}$ be any subset of edges in G that do not contain a cycle. For a graph G' either in the support of $D_{\bar{\Delta}}$ or in the support of D'_Δ, let $\sigma_{G'}(u_i, v_i) = p$ if the edge (u_i, v_i) is transformed into parallel edges in G', and let $\sigma_{G'}(u_i, v_i) = c$ if it is transformed into crossing edges. By definition of the distributions, if G' is in the support of D'_Δ, so that $G' = G_\sigma$ for $\sigma : E(G) \to \{p, c\}$, then $\sigma_{G'}(u_i, v_i) = \sigma$, and if G' is in the support of $D_{\bar{\Delta}}$, so that $G' = G_\pi$ for $\pi : V(G) \to \{\ell, r\}$, then $\sigma_{G'}(u_i, v_i) = c$ when $\pi(u_i) = \pi(v_i)$, and $\sigma_{G'}(u_i, v_i) = p$ otherwise.

Consider the distribution over $\sigma_{G'}(u_1, v_1), \ldots, \sigma_{G'}(u_t, v_t)$ when G' is chosen according to each of the two distributions. Then we claim that the distribution is uniform over $\{p, c\}^t$. This is clearly the case when G' is selected according to D'_Δ (as G' is determined by selecting $\sigma : E(G) \to \{p, c\}$ uniformly at random). It remains to verify that this is also true when $G' = G_\pi$, and $\pi : V(G) \to \{\ell, r\}$ is selected uniformly at random. Consider any vector $h \in \{p, c\}^t$ (where $h_i = p$ means that (u_i, v_i) is transformed into a parallel edge, and $h_i = c$ means that it is transformed into a crossing edge). Since F does not contain a cycle, it consists of a forest of trees, $T_1, \ldots, T_s$. The total number of vertices in these trees is $t + s$. For each tree T_j, if the value of $\pi(v)$ is determined for some arbitrary vertex v in the tree, then the setting of h enforces a unique value $\pi(u)$ for every $u \in T_j$. In other words, the number of settings of π that induce a particular h is 2^s. Since the total number of settings of π is 2^{s+t} we get:

$$\forall h \in \{p, c\}^t\,, \qquad \Pr_\pi\left[\sigma_{G_\pi}(u_1, v_1), \ldots, \sigma_{G_\pi}(u_t, v_t) = h\right] \;=\; \frac{2^s}{2^{t+s}} = \frac{1}{2^t} \tag{48}$$

and so the (modified) Independence Observation is established. The proof is completed analogously to the way it was completed under Condition 1 (using the modified Independence Observation and replacing detection of triangles with detection of cycles). ■

7 Upper Bounds

7.1 An upper bound of $\tilde{O}(\sqrt{nd}/\epsilon^{3/2})$ for general graphs

Lemma 14 *It is possible to test triangle-freeness by performing $\tilde{O}(\sqrt{nd}/\epsilon^{3/2})$ queries. If $d_{\max} = O(d)$ then $O(d/\epsilon)$ queries suffice.*

Proof: Let G be a graph with average degree d over n vertices that is ϵ-far from being triangle-free. By definition, G must contain at least ϵdn edges that belong to triangles. If $d_{\max} = O(d)$ then by uniformly selecting $\Theta(1/\epsilon)$ vertices and for each uniformly selecting an incident edge, with high probability we obtain an edge that belongs to a triangle. Conditioned on this event, if we now perform all $O(d)$ neighbor queries to the end-points of each selected edge, we reveal a triangle.

If the maximum degree of the graph differs significantly from its average degree, then the above argument cannot be applied: First, the suggested edge selection process might not select with sufficiently high probability an edge that belongs to a triangle because the algorithm uniformly selects vertices rather than edges. Second, even if we obtain such an edge, its end-points might have a very high degree. To address these issues, we first introduce some notation.

We say that a vertex has *high degree* if its degree is more than $c\sqrt{nd}$ (where we set c momentarily). We shall say that an edge is *covered* by these high degree vertices, if *both* its end-points have high degree. By definition, the high-degree vertices can cover at most $((1/c)\sqrt{nd})^2 = (1/c^2)nd$ edges. Hence, among the edges that belong to triangles, there are at least $(\epsilon - (1/c^2))nd$ edges that have at least one end-point with degree at most $c\sqrt{nd}$. If we set $c = \sqrt{2/\epsilon}$ then we have at least $(\epsilon/2)nd$ such edges.

In order to obtain one of these edges, we would like to sample edges uniformly in G. In fact, it suffices to sample edges "almost uniformly" as defined in [KKR04]. In [KKR04] an algorithm is described that uses $\tilde{O}(\sqrt{n/\delta})$ queries to a graph G and for which the following holds: For all but at most $\delta/4$-fraction of the edges of G the probability that the edge is selected is at least $\frac{1}{32nd}$. We refer to this algorithm as "Edge-Select". By definition of the algorithm, if we set $\delta = \epsilon$, we get that there are at least $(\epsilon/4)nd$ edges that can be returned by "Edge-Select" such that these edges belong to triangles and have at least one end-point with degree at most $\sqrt{2/\epsilon}\sqrt{nd}$. It follows that at a cost of $\tilde{O}(\sqrt{n}/\epsilon^{3/2})$ queries we obtain such an edge with a high constant probability. Thus the algorithm for detecting a triangle runs "Edge-Select" $\Theta(1/\epsilon)$ times. For each selected edge, if it has one end-point with degree less than $\sqrt{2/\epsilon} \cdot \sqrt{nd}$ then it asks all neighbor queries for that vertex, and for each of them it asks all pair queries with the other end point. (If both end-points have high degree then the algorithm does nothing). ■

7.2 An improved upper bound for relatively dense general graphs

Lemma 15 *It is possible to test triangle-freeness by performing $O\left(\max\left\{\frac{n^{4/3}}{\epsilon^{2/3}d^{2/3}}, \frac{d_{\max}^2}{\epsilon^2 d^2}\right\}\right)$ queries.*

As a corollary we get:

Corollary 16 *It is possible to test triangle-freeness of graphs with average degree $d = \Omega(\sqrt{n})$ by performing $O\left(\frac{n^{4/3}}{d^{2/3}\epsilon^2}\right)$ queries.*

Proof of Lemma 15. Let G be a graph over n vertices with average degree d and maximum degree $d_{\max}$ that is ϵ-far from being triangle-free. We shall show that if we take a uniform sample of $\Theta\left(\max\left\{\frac{n^{2/3}}{\epsilon^{1/3}d^{1/3}}, \frac{d_{\max}}{\epsilon d}\right\}\right)$ vertices of G, and ask vertex-pair queries between all pairs in the sample, then a triangle is detected with probability at least $2/3$.

Since G is ϵ-far from being triangle-free, it must contain at least ϵdn triples of vertices that form a triangle. This lower bound on the number of triangles implies that the expected number of triangles in a set of s uniformly selected vertices is at least $s^3 \cdot \frac{\epsilon dn}{n^3}$. It follows that for $s \geq n^{2/3}/(\epsilon d)^{1/3}$, the expected number of triangles spanned by the sample is at least 1. This unfortunately does not imply in general that a uniform sample of $s = \Omega(n^{2/3}/(\epsilon d)^{1/3})$ vertices spans a triangle with probability at least $2/3$. Rather, the size of the sample should depend on the ratio between $d_{\max}$ and d.

Let $s = c \cdot \max\left(\frac{n^{2/3}}{(\epsilon d)^{1/3}}, \frac{d_{\max}}{\epsilon d}\right)$, where $c > 0$ is a sufficiently large constant. Since G is ϵ-far from being triangle-free, G must contain a family T of $(\epsilon dn)/3$ pairwise edge-disjoint triangles. Fix such a family, and for every $v \in V(G)$, let $d_T(v)$ be the number of triangles in T containing v; obviously, $d_T(v) \leq d(v)/2 \leq d_{\max}/2$. We sample a set S of s vertices of G uniformly at random. Let X be the random variable counting the number of triangles of T spanned by S. Due to the Chebyshev inequality, it is enough to prove that $\mathrm{Exp}[X]$ is at least a large constant, and the ratio $\mathrm{Var}[X]/\mathrm{Exp}^2[X]$ is at most a small enough constant. We will estimate both quantities.

Observe that each triangle of T falls into S with probability $(1 + o(1))s^3/n^3$. It follows that

$$\mathrm{Exp}[X] = (1 + o(1))\frac{s^3}{n^3}|T| = \Theta\left(\frac{\epsilon d s^3}{n^2}\right). \tag{49}$$

Thus, taking c large enough, we get: $\mathrm{Exp}[X]$ is large enough, too. Also,

$$\begin{aligned}\mathrm{Var}[X] &\leq \sum_{\substack{t,t' \in T \\ t \cap t' \neq \emptyset}} \Pr[t, t' \subset S] \\ &= \sum_{v \in V(G)} \binom{d_T(v)}{2} \frac{(1 + o(1))s^5}{n^5}\end{aligned} \tag{50}$$

(the latter estimate is due to the fact that, since T is pairwise edge-disjoint, for any $t, t' \in T$ with $t \cap t' \neq \emptyset$, the union $t \cup t'$ contains exactly five vertices). Recall that $d_T(v) \leq d_{\max}$. Due to convexity, we get:

$$\mathrm{Var}[X] = O\left(\frac{\epsilon dn}{d_{\max}} \cdot d_{\max}^2\right)\frac{s^5}{n^5}. \tag{51}$$

Using the assumption $s = \Omega\left(\frac{d_{\max}}{\epsilon d}\right)$, we derive: $\mathrm{Var}[X]/\mathrm{Exp}^2[X]$ is small enough, as required. ■

8 Bounds on the Edge Density of Random Cayley Graphs with Large Sets of Generators

In this section we consider Random Cayley graphs. Behrend graphs are essentially variants of Cayley graphs. Using methods that were introduced in Subsection 5.2, we get bounds on the edge

density of random Cayley graphs with large sets of generators. We start with a definition of Cayley graphs. For simplicity we consider only Cayley graphs over Abelian groups, but all arguments apply to the non-Abelian case as well.

Let H be a finite Abelian group. A set $X \subseteq H$ is *symmetric* if $X = -X$, where $-X = \{-x : x \in X\}$. The *Cayley graph* over H with respect to a symmetric set X, denoted CG_X, has H as its vertex set and distinct vertices $a, b \in H$ are connected by an edge if and only if $a - b$ (hence also $b - a$) is in X. We shall say in such a case that the edge *corresponds* to $x = a - b$. All operations involving vertices are performed in H. A *difference* of a set $T \subseteq H$ is an element $a - b$ such that $a, b \in T$. We let $\Delta(T)$ denote the set $\{a - b : a, b \in T\}$ of differences of T. By definition $\Delta(T)$ contains at most $|T|(|T| - 1)$ nonzero elements. The *multiplicity* in T of a difference $x \in \Delta(T)$ is $|\{(a, b) : a, b \in T \text{ and } a - b = x\}|$. Clearly, the differences that correspond to edges that are incident to a specific vertex are all distinct. It follows that the multiplicity in T of any specific difference in $\Delta(T)$ is at most $|T|$. Moreover, if we consider the complete graph over H, then the edges that correspond to a specific difference form a set of cycles that cover H.

In what follows we show that for dense random Cayley graphs the edge density in relatively large induced subgraphs is close to the edge density of the whole graph. It was previously shown [AR94] that random Cayley graphs are expanders and hence have the property that the density of every induced subgraph on sufficiently many vertices is very close to the density of the graph. However, the known techniques for proving this property are based on estimating the second eigenvalue of the graph's adjacency matrix, and do not supply any informative bounds for sets of vertices that are much smaller than the number of vertices divided by the square root of the degree.

We shall use the following notation: For $X, C \subseteq H$, we let $e_X(C)$ denote the number of edges in the subgraph of CG_X that is induced by C.

Theorem 4 *Let $0 < \beta < \frac{1}{2}$ and $0 < \alpha \leq 1$ satisfy $\alpha - 2\beta > \frac{1}{\log \log n}$, and let H be an Abelian group where $|H| = n$. Let $X \subset H$, $X = -X$, be determined as follows: Every pair $x, -x$ is chosen independently with probability $p(n) = \frac{1}{n^\beta}$ to be in X. With high probability over the choice of X, for every subset C of n^α vertices in CG_X, we have that $e_X(C) \leq \frac{90}{\alpha - 2\beta} \frac{n^{2\alpha}}{n^\beta}$.*

Similarly to Lemma 8, Theorem 4 shows that for sufficiently large subsets C (i.e., for $|C| = n^\alpha$, where $\alpha - 2\beta$ is a constant), the number of edges spanned by C in CG_X is close to its expected value. Here too, the smaller we choose β (i.e., the larger we choose X), the smaller can α be. That is, the theorem can be applied to sets with smaller size. The proof of Theorem 4 is similar to the proof of Lemma 8. The difference lies in the proof of Claim 18, which is analogous to Claim 9. Here too we let $k = \frac{5}{\alpha - 2\beta}$ and note that for β and α as required in the theorem, $k \leq 5 \log \log n$. We shall say that W is *good* if no difference in $\Delta(W)$ has multiplicity higher than k in W. Theorem 4 follows from the next two claims using the same arguments that were applied in showing that Lemma 8 follows from Claims 9 and 10.

Claim 17 *With high probability over the choice of X, for every good W such that $|W| = s \geq n^\beta \log n$, we have that $e_X(W) \leq \frac{2ks^2}{n^\beta}$.*

Proof: Consider a fixed choice of a good W such that $|W| = s = n^\beta \log n$. By definition, $|\Delta(W)| \leq |W|^2 = s^2$. Since W is good, for every $x \in \Delta(W)$, if $x \in X$, then the number of edges

labeled by x in the subgraph of CG_X induced by W is at most k. Since each $x \in \Delta(W)$ is chosen to be in X independently with probability $n^{-\beta}$, the expected size of $\Delta(W) \cap X$ is $|\Delta(W)| \cdot n^{-\beta} \leq s^2 \cdot n^{-\beta}$. By a multiplicative Chernoff bound, the probability that $|\Delta(W)| \cap X > 2 \cdot s^2 \cdot n^{-\beta}$ is upper bounded by $\exp\left(-s^2 \cdot n^{-\beta}\right)$. The claim follows by taking a union bound over all choices of W of size s. ■

Claim 18 *Let $C \subset H$ satisfy $|C| = n^{\alpha}$. Suppose we uniformly and independently select $W \subset C$, $|W| = n^{\beta} \log n$. Then the probability that W is* not good *is at most $\frac{1}{n^{\beta}}$.*

Proof: Consider any fixed difference $x \in \Delta(C)$, where there are at most $|C|^2$ such differences. Recall that there are at most $|C|$ edges in the subgraph of CG_H induced by C that correspond to x. We denote this set of edges by $E_x(C)$. Since $E_x(H)$ is a union of disjoint cycles, $E_x(C)$ is a union of disjoint cycles and paths. Therefore, every choice of $k+1$ edges in $E_x(C)$ is a union of r_1 (sub)paths and r_2 cycles where $1 \leq r_1 \leq k+1$ and $r_2 \leq (k+1-r_1)/3$ (the second inequality follows from the fact that the length of a cycle is at least 3). Note that when $r_1 = k+1$ then the $k+1$ edges constitute a matching, as was the case in the proof of Claim 10. In this case the number of vertices incident to them is $2(k+1)$.

More generally, the number of incident vertices is $k+1+r_1$. For each pair (r_1, r_2), the number of choices of $k+1$ edges that constitute r_1 paths and r_2 cycles is at most $|C|^{r_1}(k+1)^{r_1} \cdot |C|^{r_2}$. Namely, to determine each of the r_1 paths, we select a starting vertex (out of $|C|$ vertices) and a length (between 1 and $k+1$). To determine each of the r_2 cycles, we select a vertex (that belongs to the cycle). Once the edges are selected, the number of choices of the remaining $|W|-(k+1+r_1)$ vertices in W is $\binom{|C|-(k+1+r_1)}{|W|-(k+1+r_1)}$. Therefore, for each fixed choice of r_1 and r_2, the probability that W spans $k+1$ edges in $E_x(C)$ that constitute r_1 paths and r_2 cycles is at most

$$\begin{aligned}
\frac{|C|^{r_1}(k+1)^{r_1}|C|^{r_2}\binom{|C|-(k+1+r_1)}{|W|-(k+1+r_1)}}{\binom{|C|}{|W|}} &\leq |C|^{r_1}(k+1)^{r_1}|C|^{r_2} \cdot \left(\frac{|W|}{|C|}\right)^{k+1+r_1} && (52)\\
&= (k+1)^{r_1} \cdot \frac{|W|^{k+1+r_1}}{|C|^{k+1-r_2}} && (53)\\
&< (k+1)^{r_1} \frac{|W|^{k+1+r_1}}{|C|^{(2/3)(k+1)+(1/3)r_1}} && (54)
\end{aligned}$$

The expression in Equation (54) is maximized when the ratio between $|W|$ and $|C|$ is maximized (i.e., β is maximized with respect to α) and r_1 is maximized (i.e., $r_1 = k+1$). In this case we get an upper bound of $(k+1)^{k+1}(\log n)^{2(k+1)} n^{(2\beta-\alpha)(k+1)}$. Substituting $k = \frac{5}{\alpha-2\beta}$, where $k \leq 5 \log\log n$, summing over all r_1, r_2 and taking a union bound over the at most $|C|^2 = n^{2\alpha} \leq n^2$ choices of $x \in \Delta(C)$, the claim follows. ■

It is worth noting that the technique here can be applied with other parameters as well. In particular, it can be shown, for example, that with high probability, in random Cayley graphs over groups of size n in which the number of generators is $(1+o(1))\frac{n}{2}$, every set X of at least some $\mathrm{poly}\log(n)$ vertices spans $(1+o(1))\frac{1}{2}\binom{|X|}{2}$ edges. This is related to the results in [AO95] and [Gre05].

Acknowledgement. The authors wish to thank Angelika Steger for bringing theses [Gug06] and [Ras06] to their attention.

References

[AFKS00] N. Alon, E. Fischer, M. Krivelevich, and M Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000.

[Alo02] N. Alon. Testing subgraphs in large graphs. *Random Structures and Algorithms*, 21(3–4):359–370, 2002.

[AO95] N. Alon and A. Orlitsky. Repeated communication and ramsey graphs. *IEEE Transactions on Information Theory*, 41:1276–1289, 1995.

[AR94] N. Alon and Y. Roichman. Random Cayley graphs and expanders. *Random Structures and Algorithms*, 5:271–284, 1994.

[AS04a] N. Alon and A. Shapira. A characterization of easily testable induced subgraphs. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004.

[AS04b] N. Alon and A. Shapira. Testing subgraphs in directed graphs. *JCSS*, 69:354–482, 2004.

[Beh46] F. A. Behrend. On sets of integers which contain no three terms in arithmetic progression. *Proc. National Academy of Sciences USA*, 32:331–332, 1946.

[Fis01] E. Fischer. The art of uninformed decisions: A primer to property testing. *The Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *JACM*, 45(4):653–750, 1998.

[GR02] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

[Gre05] B. Green. Counting sets with small sumset, and the clique number of random Cayley graphs. *Combinatorica*, 25(3):307–326, 2005.

[Gug06] L. Gugelmann. Testing trinagle-freeness in general graphs: Lower bounds, 2006. Bachelor thesis, Dept. of Mathematics, ETH, Zurich.

[JuR00] S. Janson, T. Łuczak, and A. Ruciński. *Random graphs*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[KKR04] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1441–1483, 2004.

[PR02] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, 20(2):165–183, 2002.

[Ras06] T. Rast. Testing trinagle-freeness in general graphs: Upper bounds, 2006. Bachelor thesis, Dept. of Mathematics, ETH, Zurich.

[Ron01] D. Ron. Property testing. In *Handbook of Randomized Computing*, Volume II, Chapter 15, pages 597–649. Edited by S. Rajasekaran, P. M. Pardalos, J.H. Reif and J. Rolim, Kluwer Academic Publishers, 2001.

[RS76] I. Z. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely), Coll. Math. Soc. J. Bolyai 18*, 2:939–945, 1976.

[RS96] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[SS42] R. Salem and D. C. Spencer. On sets of integers which contain no three terms in arithmetical progression. *Proc. National Academy of Sciences USA*, 28:561–563, 1942.

A Proof of Claim 4

Recall that this claim is part of the proof of Lemma 3. Specifically, it is part of the probabilistic analysis of the number of disjoint triangles obtained when drawing a graph according to the distribution D_Δ, which is defined as follows. A graph is generated by first partitioning the vertices into equal-size subsets of size $n' = n/3$, denoted V_1, V_2, V_3, where the vertices in each V_ℓ, $\ell \in \{1,2,3\}$ are denoted $\{v_{\ell,1}, \ldots, v_{\ell,n'}\}$. Next, between each pair of subsets, $d' = d/2 = \sqrt{n}/3$ random perfect matchings are selected. Recall that $\eta_{i,j,k}$ is a 0/1 random variable that equals 1 if and only if the graph contains the triangle $(v_{1,i}, v_{2,j}, v_{3,k})$ and there is no other triangle that shares an edge with this triangle. The claim is that for $i' \neq i$, $j' \neq j$, $k' \neq k$

$$\Pr[\eta_{i',j',k'} = 1 \mid \eta_{i,j,k} = 1] \;=\; \left(1 + O\left(\frac{1}{d'}\right)\right) \cdot \Pr[\eta_{i',j',k'} = 1] \tag{55}$$

To prove the claim we break both $\Pr[\eta_{i',j',k'} = 1]$ and $\Pr[\eta_{i',j',k'} = 1 \mid \eta_{i,j,k} = 1]$ into a product of (conditional) probabilities, and bound the ratio between each corresponding pair of probabilities. Recall that $\alpha_{i,j}^{\ell,\ell'}$ is a 0/1 random variable that indicates the existence of the edge $(v_{\ell,i}, v_{\ell',j})$, $\Delta_{i,j,k}$ is a 0/1 random variable that indicates the existence of the triangle $(v_{1,i}, v_{2,j}, v_{3,k})$, and $\beta_{i,j,k}$ is a 0/1 random variable that indicates whether there is a triangle different from $(v_{1,i}, v_{2,j}, v_{3,k})$ that contains one of the three edges: $(v_{1,i}, v_{2,j})$, $(v_{1,i}, v_{3,k})$, $(v_{2,j}, v_{3,k})$. By definition,

$$\Pr[\eta_{i',j',k'} = 1] \;=\; \Pr[\Delta_{i',j',k'} = 1] \cdot \Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1] \tag{56}$$

and

$$\Pr[\eta_{i',j',k'} = 1 \mid \eta_{i,j,k} = 1] \;=\; \Pr[\Delta_{i',j',k'} = 1 \mid \eta_{i,j,k} = 1] \cdot \Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \eta_{i,j,k} = 1] \tag{57}$$

We first bound the ratio between $\Pr[\Delta_{i',j',k'} = 1 \mid \eta_{i,j,k} = 1]$ and $\Pr[\Delta_{i',j',k'} = 1]$, and then between $\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \eta_{i,j,k} = 1]$ and $\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1]$.

Before doing so we make a few observations (some of which were already used in the proof of Lemma 3). For illustrations of these observations see Figure 3. Consider an event of the form $\alpha_{r,s}^{\ell,\ell'} = 1$. We know that $\Pr\left[\alpha_{r,s}^{\ell,\ell'} = 1\right] = 1 - \left(1 - \frac{1}{n'}\right)^{d'}$. How is the probability of this event influenced by the existence or non-existence of other edges? Consider first conditioning on the

event that $\alpha^{\ell,\ell'}_{r',s'} = 1$ for $r' \neq r$ and $s' \neq s$. This means that for at least one of the d' matchings between V_ℓ and $V_{\ell'}$, the vertex s' cannot be matched to r (and similarly, r' cannot be matched to s). In other words, in each of these matchings where the vertex s' cannot be matched to r, the probability that r is matched to s is $\frac{1}{n'-1}$ rather than $\frac{1}{n'}$. Therefore, for $r' \neq r$, $s' \neq s$,

$$\Pr\left[\alpha^{\ell,\ell'}_{r,s} = 1 \mid \alpha^{\ell,\ell'}_{r',s'} = 1\right] \;\leq\; 1 - \left(1 - \frac{1}{n'-1}\right)^{d'} \;\leq\; \frac{d'}{n'-1} = \frac{d'}{n'} \cdot \frac{n'}{n'-1} \tag{58}$$

That is, the probability that $\alpha^{\ell,\ell'}_{r,s} = 1$ slightly increases when conditioning on $\alpha^{\ell,\ell'}_{r',s'} = 1$. Next, consider conditioning on there *not* being an edge between r' and s for $r' \neq r$, that is $\alpha^{\ell,\ell'}_{r',s} = 0$. We claim that the probability that $\alpha^{\ell,\ell'}_{r,s} = 1$ increases as well in this case. This is true because $\alpha^{\ell,\ell'}_{r',s} = 0$ means that in each of the d' matchings, r' is matched to some vertex $s' \neq s$, so that the conditional probability that r is matched to s in every matching is $\frac{1}{n'-1}$. Therefore, for $r' \neq r$,

$$\Pr\left[\alpha^{\ell,\ell'}_{r,s} = 1 \mid \alpha^{\ell,\ell'}_{r',s} = 0\right] \;=\; 1 - \left(1 - \frac{1}{n'-1}\right)^{d'} \;\leq\; \frac{d'}{n'-1} = \frac{d'}{n'} \cdot \frac{n'}{n'-1} \tag{59}$$

Finally we observe that for any $s' \neq s$

$$\Pr\left[\alpha^{\ell,\ell'}_{r,s} = 1 \mid \alpha^{\ell,\ell'}_{r',s} = 0 \;\&\; \alpha^{\ell,\ell'}_{r',s'} = 1\right] = \Pr\left[\alpha^{\ell,\ell'}_{r,s} = 1 \mid \alpha^{\ell,\ell'}_{r',s} = 0\right] \tag{60}$$

This is true since the event $\alpha^{\ell,\ell'}_{r',s} = 0$ implies that $\alpha^{\ell,\ell'}_{r',s'} = 1$ for some $s' \neq s$, and the identity of s' is irrelevant when computing the probability that $\alpha^{\ell,\ell'}_{r,s} = 1$.

Given the above discussion, how does the conditioning on $\eta_{i,j,k} = 1$ (i.e., $\Delta_{i,j,k} = 1$ and $\beta_{i,j,k} = 0$) influence the probability of the event $\Delta_{i',j',k'} = 1$? Consider the event $\alpha^{1,2}_{i',j'} = 1$ (the other two events, $\alpha^{1,3}_{i',k'}$ and $\alpha^{2,3}_{j',k'}$, are analyzed similarly). Since $\Delta_{i,j,k} = 1$, we have that $\alpha^{1,2}_{i,j} = 1$. Since $\beta_{i,j,k} = 0$ we know that for every $j'' \neq j$ either $\alpha^{1,2}_{i,j''} = 0$ or $\alpha^{2,3}_{j'',k} = 0$. In particular, setting $j'' = j'$ we have that either $\alpha^{1,2}_{i,j'} = 0$ or $\alpha^{2,3}_{j',k} = 0$. Recall that we have argued above that conditioning on $\alpha^{1,2}_{i,j'} = 0$ can only increase the probability that $\alpha^{1,2}_{i',j'} = 1$ (assuming $i' \neq i$). Therefore,

$$\begin{aligned}
\Pr\left[\alpha^{1,2}_{i',j'} = 1 \mid \eta_{i,j,k} = 1\right] &\leq \Pr\left[\alpha^{1,2}_{i',j'} = 1 \mid \alpha^{1,2}_{i,j} = 1 \;\&\; \alpha^{1,2}_{i,j'} = 0\right] && (61)\\
&\leq \Pr\left[\alpha^{1,2}_{i',j'} = 1 \mid \alpha^{1,2}_{i,j'} = 0\right] && (62)\\
&\leq \frac{d'}{n'} \cdot \frac{n'}{n'-1} && (63)\\
&\leq \frac{d'}{n'} \cdot \left(1 + \frac{2}{n'}\right) . && (64)
\end{aligned}$$

In Equation (62) we applied Equation (60), in Equation (63) we applied Equation (59), and in Equation (64) we used the assumption that $n' \geq 2$. The same upper bound holds for $\alpha^{2,3}_{j,k}$ and $\alpha^{1,3}_{i,k}$. Using the lower bound on $\Pr[\Delta_{i',j',k'} = 1]$ given in Equation (6) we get

$$\frac{\Pr[\Delta_{i',j',k'} = 1 \mid \eta_{i,j,k} = 1]}{\Pr[\Delta_{i',j',k'} = 1]} \;\leq\; \frac{\left(1 + \frac{2}{n'}\right)^3}{\left(1 - \frac{1}{18d'}\right)^3} \;\leq\; 1 + O\left(\frac{1}{d'}\right) \tag{65}$$

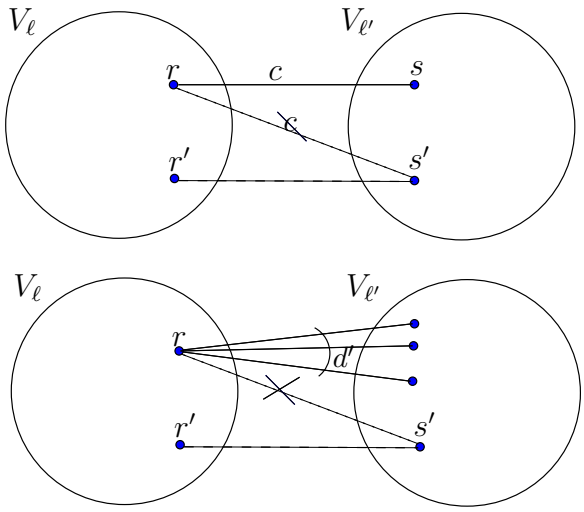

Figure 3: An illustration of how the existence and non-existence of edges influences the probability of obtaining another edge. On the top is an illustration of the case in which there is an edge between $v_{\ell,r}$ and $v_{\ell',s}$, and the event we are interested in is the existence of an edge between $v_{\ell,r'}$ and $v_{\ell',s'}$ where $r' \neq r$ and $s' \neq s$. In this illustration the label of the edge between $v_{\ell,r}$ and $v_{\ell',s}$ (the number of the matching among the d' matchings) is c, which implies that in this matching, $v_{\ell,r}$ cannot be matched to $v_{\ell',s'}$, which increases the probability that $v_{\ell,r'}$ is matched to $v_{\ell',s'}$ in this matching. On the bottom is an illustration of the case that there is no edge between $v_{\ell,r}$ and $v_{\ell',s'}$. This means that in all d' matchings, $v_{\ell,r}$ is matched to other vertices in $V_{\ell'}$, which increases the probability, in every matching, that $v_{\ell,r'}$ is matched to $v_{\ell',s'}$.

It remains to upper bound the ratio between $\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \eta_{i,j,k} = 1]$ and $\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1]$. We start by observing that since the constraints imposed by the conditioning on $\beta_{i,j,k} = 0$ can only increase the probability of *having* a triangle that shares an edge with $(v_{1,i'}, v_{2,j'}, v_{3,k'})$,

$$\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \eta_{i,j,k} = 1] \;\leq\; \Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \Delta_{i,j,k} = 1] \tag{66}$$

Next, in order to upper bound the ratio between $\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \Delta_{i,j,k} = 1]$ and $\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1]$, we lower bound the ratio between $\Pr[\beta_{i',j',k'} = 1 \mid \Delta_{i',j',k'} = 1 \;\&\; \Delta_{i,j,k} = 1]$ and $\Pr[\beta_{i',j',k'} = 1 \mid \Delta_{i',j',k'} = 1]$. This suffices since $\Pr[\beta_{i',j',k'} = 1 \mid \Delta_{i',j',k'} = 1] < 1/2$, and for every $y \leq 1/2$, if we have that $x \geq (1-\alpha)y$ for some $\alpha < 1$, then $(1-x) \geq (1+\alpha)(1-y)$.

The event $\beta_{i',j',k'} = 1$ is a union of events of the form $\Delta_{i',j',k''} = 1$ for $k'' \neq k'$ (the other two types of events, $\Delta_{i',j'',k'} = 1$ and $\Delta_{i'',j',k'} = 1$ are analyzed analogously.) The key observation is that for every $k'' \notin \{k', k\}$, the conditioning on the event $\Delta_{i,j,k} = 1$ (in addition to the conditioning on the event $\Delta_{i',j',k'} = 1$) can only increase the probability that $\Delta_{i',j',k''} = 1$ (since for at least one of the matchings, i' cannot be matched to k). Therefore, the only case in which the conditioning on $\Delta_{i,j,k} = 1$ decreases the probability that $\Delta_{i',j',k''} = 1$, is for $k'' = k$. However, the weight of the event $\Delta_{i',j',k} = 1$ (and similarly $\Delta_{i,j',k'} = 1$ and $\Delta_{i',j,k'}$) relative to $\beta_{i',j',k'} = 1$ is $O(1/n')$, and so

$$\Pr[\beta_{i',j',k'} = 1 \mid \Delta_{i',j',k'} = 1 \;\&\; \Delta_{i,j,k} = 1] \;\geq\; \left(1 - O\left(\frac{1}{n'}\right)\right) \cdot \Pr[\beta_{i',j',k'} = 1 \mid \Delta_{i',j',k'} = 1] \tag{67}$$

from which it follows that

$$\frac{\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1 \;\&\; \Delta_{i,j,k} = 1]}{\Pr[\beta_{i',j',k'} = 0 \mid \Delta_{i',j',k'} = 1]} \;\leq\; 1 + O\left(\frac{1}{n'}\right) \tag{68}$$

The claim follows by combining Equations (56), (57), (65), (66), and (68)