Electronic Colloquium on Computational Complexity, Report No. 46 (2015)

Approximately Counting Triangles in Sublinear Time

(Full Version)

Talya Eden* Amit Levi† Dana Ron‡

Abstract

We consider the problem of estimating the number of triangles in a graph. This problem has been extensively studied in two models: Exact counting algorithms, which require reading the entire graph, and streaming algorithms, where the edges are given in a stream and the memory is limited. In this work we design a *sublinear-time* algorithm for approximating the number of triangles in a graph, where the algorithm is given query access to the graph. The allowed queries are degree queries, vertex-pair queries and neighbor queries.

We show that for any given approximation parameter $0 < \epsilon < 1$, the algorithm provides an estimate $\widehat{\Delta}$ such that with high constant probability, $(1-\epsilon)\Delta(G) < \widehat{\Delta} < (1+\epsilon)\Delta(G)$, where $\Delta(G)$ is the number of triangles in the graph G. The expected query complexity of the algorithm is $O\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$, where n is the number of vertices in the graph and m is the number of edges, and the expected running time is $O\left(\frac{n}{\Delta(G)^{1/3}} + \frac{m^{3/2}}{\Delta(G)}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$. We also prove that $\Omega\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right)$ queries are necessary, thus establishing that the query complexity of this algorithm is optimal up to polylogarithmic factors in n (and the dependence on $1/\epsilon$).

*School of Computer Science, Tel Aviv University
†School Electrical Engineering, Tel Aviv University
‡School Electrical Engineering, Tel Aviv University

ISSN 1433-8092

Contents

1 Introduction

We consider the problem of approximating the number of triangles in a graph. Detecting and counting small subgraphs in a graph is a core problem in Computer Science, motivated by applications in a variety of areas, as well as by the basic quest to understand simple structures in graphs. Specifically, a triangle is one of the most elementary of such structures. The number of triangles in a graph is an important metric in a range of research areas including the study of Social Networks, Computer Communication, Bioinformatics and more (e.g., see [Col88, Was94, Por00, MSOI+02, EM02, FWVDC10]).

Counting the number of triangles in a graph G with n vertices and m edges can be performed in a straightforward manner in time $O(n^3)$, given access to the adjacency matrix of G, and in time $O(m \cdot n)$, given also access to the incidence-lists representation of G. This has been improved upon in a sequence of works [IR78, CN85, AYZ97, Sch07, Lat08, Avr10, CC11, SPK14], where the fastest known exact counting algorithm is by Alon et al. [AYZ97]. Their algorithm is based on fast matrix multiplication and runs in time $O(m^{\frac{2\omega}{\omega+1}})$ where $\omega < 2.376$ is the exponent of matrix multiplication.[1]

The problem of counting triangles has also been extensively studied in the streaming model, where the edges are given to the algorithm in a stream and the goal is to output an approximation of the number of triangles, while keeping the space complexity minimal. We give further details regarding previous works in the streaming model in Subsection 1.3.2.

Both models described above require reading the entire graph, which is not feasible in many applications. In this work, we consider a different model, in which there is query access to the graph, and the goal is to obtain an approximation of the number of triangles by performing a number of queries that is sublinear in the size of the graph. As we shall discuss later in detail, there is a growing body of works dealing with approximating graph parameters in sublinear time. There are three types of standard queries that have been considered in previous works: (1) Degree queries, in which the algorithm can query the degree $d(v)$ of any vertex v. (2) Neighbor queries, in which the algorithm can query what vertex is the i^{th} neighbor of a vertex v, for any $i \leq d(v)$. (3) Vertex-pair queries, in which the algorithm can query for any pair of vertices v and u whether (u, v) is an edge.

Gonen et al. [GRS11], who studied the problem of approximating the number of stars in a graph in sublinear time, also considered the problem of approximating the number of triangles in sublinear time. They proved that there is no sublinear approximation algorithm for the number of triangles when the algorithm is allowed to perform degree and neighbor queries (but not pair queries).[2] They raised the natural question whether such an algorithm exists when allowed vertex-pair queries in addition to degree and neighbor queries. We show that this is indeed the case, as explained next.

[1] This upper bound is given in [AYZ97] for the triangle-finding problem, but, as observed by Latapy [Lat08] it also holds for triangle counting (when given access to both the adjacency matrix and the incidence-lists representations of G).

[2] To be precise, they showed that there exist two families of graphs over $m = \Theta(n)$ edges, such that all graphs in one family have $\Theta(n)$ triangles, all graphs in the other family have no triangles, but in order to distinguish between a random graph in the first family and random graph in the second family, it is necessary to perform $\Omega(n)$ degree and neighbor queries.

1.1 Results

We describe an algorithm that, given an approximation parameter $0 < \epsilon < 1$ and query access to a graph G, outputs an estimate $\widehat{\Delta}$, such that with high constant probability (over the randomness of the algorithm), $(1-\epsilon)\Delta(G) \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G)$. The expected query complexity of the algorithm is

$$O\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon) \ ,$$

and its expected running time is $O\left(\frac{n}{\Delta(G)^{1/3}} + \frac{m^{3/2}}{\Delta(G)}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$. We show that this result is almost optimal by proving that the number of queries performed by any multiplicative-approximation algorithm for the number of triangles in a graph is

$$\Omega\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right) \ .$$

1.2 Ideas and techniques

1.2.1 The algorithm

In what follows we assume that the algorithm has some initial constant factor estimates, $\overline{m}$ and $\overline{\Delta}$, of the number of edges and triangles in the graph, respectively. Namely, $\overline{m} \geq \frac{m}{c_m}$ and $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq c_\Delta \cdot \Delta$ for constants $c_m \geq 1$ and $c_\Delta \geq 1$. This assumption can be removed by performing a combined (careful) search for such estimates.

Our starting point is similar to the ones in [GR08] and [GRS11]. We consider a partition of the graph's vertices into $O(\log n/\epsilon)$ buckets, denoted $B_0, \ldots, B_k$. In each bucket all the vertices have the same triangles-degree up to a multiplicative factor of $(1 \pm \beta)$, where β is $\Theta(\epsilon)$. The triangles-degree of a vertex v is the number of triangles that v participates in. If we could get a good estimate of the number of vertices in each bucket, then we would have a good estimate of the total number of triangles in the graph. This raises two difficulties. The first is that some of the buckets might be too small to even be "hit" by a sublinear number of samples. The second difficulty is that once we sample a vertex, we need some means by which to determine to which bucket the vertex belongs to.

In order to address the second issue we present a procedure named Approx-Triangles-Degree, which roughly does the following. Given a vertex v and an index j, the procedure determines (approximately) whether v belongs to B_j. The query complexity of the procedure is (roughly) $O\left(\frac{d(v)\sqrt{m}}{(1+\beta)^j}\right)$, where $d(v)$ is the (neighbor) degree of the vertex v. In order to achieve this complexity, the procedure handles differently vertices whose degree is at most $\sqrt{m}$ (referred to as low-degree vertices), and vertices whose degree is more than $\sqrt{m}$ (referred to as high-degree vertices). For low-degree vertices the procedure simply samples pairs of neighbors of v and performs pair queries on them. For high-degree vertices such a sampling process does not achieve the desired query complexity, and hence a more sophisticated process is applied (for more details see Subsection 3.3.2).

Returning to the first issue (of hitting small buckets), how small should a bucket B_j be so that it can be disregarded (i.e., assumed to be empty)? First observe that total number of possible triangles in which all three endpoints reside in buckets of size at most $\frac{(\beta\overline{\Delta})^{1/3}}{k+1}$ (each) is at most $\beta\overline{\Delta}$ (recall that there are $k+1$ buckets), and hence such buckets can be disregarded. Furthermore, if

the number of triangles that have at least one endpoint in a bucket B_j, that is, $|B_j|(1+\beta)^j$, is at most $\frac{\beta\overline{\Delta}}{k+1}$, then the bucket can be disregarded as well, since its contribution to the total number of triangles is not significant. This implies that as j increases, the size of B_j may be smaller, while it still has a significant contribution to the total number of triangles. Therefore, the cost of estimating $|B_j|$ for such a "significant" bucket B_j, may grow with $(1+\beta)^j$.

Recall that the complexity of Approx-Triangles-Degree decreases as the index j increases, and so we obtain a useful tradeoff between the size of the sample necessary to "hit" a bucket B_j and the complexity of determining whether a vertex v belongs to B_j. This tradeoff actually eliminates the dependence on $(1+\beta)^j$. However, the complexity of Approx-Triangles-Degree also has a linear dependence on $d(v)$, which may be large. Here we observe another tradeoff: While the complexity of the procedure increases with $d(v)$, the number of vertices with at least a given degree d is upper bounded by $2m/d$, which decreases with d.

By exploiting both tradeoffs we can show that it is possible to obtain an estimate $\widehat{\Delta}$ such that (with high probability) $(\frac{1}{3}-\epsilon)\Delta(G) \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G)$ and the number of queries performed is $O\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$. In order to improve the result and get a $(1\pm\epsilon)$-approximation of the number of triangles we build on and extend related ideas presented in [GR08, GRS11]. In particular, we let triangles "observed" from endpoints in sufficiently large buckets (which are sampled) compensate for triangles "observed" from endpoints in small buckets (which are not sampled).

1.2.2 The lower bound

Proving that every multiplicative-approximation algorithm must perform $\Omega\left(\frac{n}{\Delta(G)^{1/3}}\right)$ queries is fairly straightforward, and our main focus is on proving that $\Omega\left(\min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right)$ queries are necessary as well. In order to prove this claim we define, for every n, every $1 \leq m \leq \binom{n}{2}$ and every $1 \leq \Delta \leq \min\{\binom{n}{3}, m^{3/2}\}$, a graph G_1 and a family of graphs $\mathcal{G}_2$ for which the following holds: (1) The graph G_1 and all the graphs in $\mathcal{G}_2$ have n vertices and m edges. (2) In G_1 there are no triangles, that is, $\Delta(G_1) = 0$, while $\Delta(G) = \Theta(\Delta)$ for every graph $G \in \mathcal{G}_2$. We prove that for values of Δ such that $\Delta \geq \sqrt{m}$, at least $\Omega\left(\frac{m^{3/2}}{\Delta}\right)$ queries are required in order to distinguish with high constant probability between G_1 and a random graph in $\mathcal{G}_2$. We then prove that for values of Δ such that $\Delta < \sqrt{m}$, at least $\Omega(m)$ queries are required for this task. We give four different constructions for G_1 and $\mathcal{G}_2$ depending on the value of Δ as a function of m.

1.3 Related Work

1.3.1 Approximating the number of subgraphs and other graph parameters in sublinear time

Our work extends the works [Fei06, GR08] on approximating the average degree of a graph (the number of edges) and the work of [GRS11] on approximating the number of stars in a graph, in sublinear time. Feige [Fei06] investigated the problem of estimating the average degree of a graph, denoted $\overline{d}$, when given query access to the degrees of the vertices. He proved that $O(\sqrt{n}/\epsilon)$ queries are sufficient in order to obtain a $(\frac{1}{2}-\epsilon)$-approximation of $\overline{d}$ (conditioned on $\overline{d} = \Omega(1)$) and proved that a better approximation ratio cannot be achieved in sublinear time using only degree queries.

The same problem was considered by Goldreich and Ron [GR08]. They proved that, when allowing neighbor queries as well as degree queries, $O(\sqrt{n}) \cdot \text{poly}(\log n, 1/\epsilon)$ queries are sufficient in order to obtain a $(1 \pm \epsilon)$-approximation of $\overline{d}$. In both results the term $\sqrt{n}$ can actually be replaced by $\sqrt{n/\overline{d}}$, so that the complexity of the algorithms improves as $\overline{d}$ increases.

Gonen et al. [GRS11] considered the problem of approximating the number of s-stars in a graph. That is, subgraphs over $s+1$ vertices, where one vertex is connected to all others. They presented an algorithm that, given an approximation parameter $0 < \epsilon < 1$ and query access to a graph G, outputs an estimate $\hat{\nu}_s$ such that with high constant probability $(1-\epsilon)\nu_s(G) \leq \hat{\nu}_s \leq (1+\epsilon)\nu_s(G)$, where $\nu_s(G)$ denotes the number of s-stars in the graph. The expected query complexity and running time of their algorithm are $O\left(\frac{n}{\nu_s(G)^{1/(s+1)}} + \min\left\{n^{1-1/s}, \frac{n^{s-1/s}}{\nu_s(G)^{1-1/s}}\right\}\right) \cdot \text{poly}(\log n, 1/\epsilon)$.

Additional works on sublinear algorithms for estimating other graph parameters include those for approximating the size of the minimum weight spanning tree [CRT05, CS09, CEF+05], maximum matching [NO08, YYI09] and of the minimum vertex cover [PR07, MR09, NO08, YYI09, HKNO09, ORRR12].

1.3.2 Counting the number of triangles in the streaming model

Bar-Yossef et al. [BYKS02] initiated the study of counting the number of triangles in the streaming model. Many works have been conducted since, e.g. [JG05, BFL+06, BBCG08, TKMF09, TDM+09, TKM11, YK11, KMPT12], differing in the number of passes they perform, the assumptions they make on the structure of the graph, the requirements on the output and more. A work with some resemblance to ours is the work of Kolountzakis et al. [KMPT12]. They present a streaming algorithm that makes three passes over the edge stream and outputs a $(1 \pm \epsilon)$-approximation of $\Delta(G)$. The space complexity of the algorithm is $O\left(\sqrt{m} \cdot \log m + \frac{m^{3/2} \cdot \log n}{\Delta(G) \cdot \epsilon^2}\right)$ (where m, n and $\Delta(G)$ are as defined previously). The point of similarity between their algorithm and ours is that they also classify the graph's vertices into high-degree vertices, with degree strictly greater than $\sqrt{m}$, and low-degree vertices, with degree at most $\sqrt{m}$, and apply a different approach to estimate the triangles-degree of each type of vertices. Since their algorithm requires several passes over the edge stream and relies heavily on direct access to uniformly selected edges, it otherwise clearly differs from our algorithm.

2 Preliminaries

Let $G = (V, E)$ be a simple graph with $|V| = n$ vertices and $|E| = m$ edges. We denote by $d(v)$ the degree of the vertex $v \in V$ and by $\Gamma(v)$ the set of v's neighbors. For a vertex v we refer to the number of triangles v participates as the triangles degree of v, and denote it by $\Delta(v)$. We denote the set of triangles that a vertex v participates in by $\text{Tr}(v)$.

All our algorithms can sample uniformly in V and perform three types of queries:

1. Degree queries, in which the algorithm may query for the degree $d(v)$ of any vertex v of its choice.

2. Neighbor queries, in which the algorithm may query for the i^{th} neighbor of any vertex v of its choice. If $i > d(v)$, then a special symbol (e.g. †) is returned. No assumption is made on the order of the neighbors of any vertex.

3. Pair queries, in which the algorithm may ask if there is an edge $(u,v) \in E$ between any pair of vertices u and v.

We denote by $\mathrm{Tr}(G)$ the set of triangles in the graph G, and by $\Delta(G)$ the number of triangles in the graph. Each triangle, that is three vertices $u,v,w \in V$ such that (u,v), (v,w) and (u,w) are edges in G, is denoted by an unordered triple (v,u,w).

We note that we sometimes use set notations for operations on multisets. Also, we start with the following assumption.

Assumption 2.1 *Our algorithms take as input estimates $\overline{\Delta}$ and $\overline{m}$ on the number of edges and triangles in the graph respectively, such that*

1. $\frac{\Delta(G)}{c_\Delta} \leq \overline{\Delta} \leq \Delta(G)$, *for some constant c_Δ that will be set later on.*
2. $\overline{m} \geq \frac{m}{c_m}$, *for some constant c_m that will be set later on.*
3. $\overline{\Delta} \leq \overline{m}^{3/2}$

Where the third assumption is justified by the following claim.

Proposition 2.2 *For every G, $\Delta(G) \leq m^{3/2}$.*

Proof: Note that for every v it holds that $\Delta(v) \leq m$ and that $\Delta(v) \leq d(v)^2$. Therefore,

$$\begin{aligned}\Delta(G) = \frac{1}{3}\sum_{v\in V}\Delta(v) &\leq \frac{1}{3}\left(\sum_{v:\, d(v)>\sqrt{m}}\Delta(v) + \sum_{v:\, d(v)\leq\sqrt{m}} d(v)^2\right) \\ &\leq \frac{1}{3}\left(2\sqrt{m}\cdot m + \sqrt{m}\sum_{v:\, d(v)\leq\sqrt{m}} d(v)\right) \leq m^{3/2},\end{aligned}$$

and the proof is complete. ■

We remove the need for this a priori knowledge on $\Delta(G)$ and m in Section 3.5.

Since we shall use the multiplicative Chernoff bound extensively, we quote it next. Let $\chi_1,\ldots,\chi_r$ be r independent random variables, such that $\chi_i \in [0,1]$ and $\Pr[\chi_i = 1] = p$ for every $1 \leq i \leq r$. For every $\gamma \in (0,1]$ the following holds:

$$\Pr\left[\frac{1}{r}\sum_{i=1}^{r}\chi_i > (1+\gamma)p\right] < \exp\left(-\gamma^2 pr/3\right), \tag{1}$$

and

$$\Pr\left[\frac{1}{r}\sum_{i=1}^{r}\chi_i < (1-\gamma)p\right] < \exp\left(-\gamma^2 pr/2\right). \tag{2}$$

Observe that Equation (1) holds also for independent random variables $\chi_1,\ldots,\chi_r$, such that for every $i \in [r]$, $\Pr[\chi_i = 1] \leq p$. Similarly Equation (2) holds also for independent random variables $\chi_1,\ldots,\chi_r$, such that for every $i \in [r]$, $\Pr[\chi_i = 1] \geq p$.

3 An algorithm for approximating the number of triangles

We shall say that an algorithm is a $\frac{1}{3}$-approximation algorithm for the number of triangles if, for any graph G, given as input $0 < \epsilon < 1$, the algorithm computes an estimate $\widehat{\Delta}$ such that with high constant success probability $\frac{1}{3}(1-\epsilon)\Delta(G) \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G)$. In Subsection 3.2 we present a $\frac{1}{3}$-approximation algorithm for the number of triangles assuming we have query access to the triangles-degree $\Delta(v)$ of each vertex v of our choice. In Subsections 3.3–3.3.4 we show how to remove this assumption and obtain a $\frac{1}{3}$-approximation algorithm that does not have access to such an oracle. In Subsection 3.4 we show how the $\frac{1}{3}$-approximation algorithm can be modified so as to obtain a $(1 \pm \epsilon)$-approximation of the number of triangles. The aforementioned algorithms work under the assumption that they are provided with constant-factor estimates of m and $\Delta(G)$ (as defined in Assumption 2.1). In Subsection 3.5 we describe how to eliminate the need for a priori knowledge on m and $\Delta(G)$, and we analyze the complexity of the resulting algorithm and the above listed algorithms. Since the algorithm includes many details, we first provide an overview in Subsection 3.1.

3.1 An overview of the algorithm

As noted in the introduction, we consider a partition of the graph's vertices into $O(\log n/\epsilon)$ buckets, denoted $B_0, \ldots, B_k$. In each bucket B_j, all the vertices have approximately the same triangles-degree $(1+\beta)^j$, where β is $\Theta(\epsilon)$. Observe that $\frac{1}{3}\sum_j |B_j| \cdot (1+\beta)^j$ is within $(1 \pm \beta)$ of $\Delta(G)$, since each triangle is counted three times, once from each endpoint.

3.1.1 An oracle-based algorithm

We start by assuming that we have query access to an oracle that, given a vertex v, replies with the triangles-degree of v. We prove that given such an oracle, if we disregard "small" buckets and only estimate the sizes $|B_j|$ of the "large" buckets, for an appropriate threshold of largeness, then we can get an approximation $\widehat{\Delta}$ such that $(\frac{1}{3}-\epsilon)\Delta(G) \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G)$. We set the threshold of largeness at $\frac{(\beta\overline{\Delta})^{1/3}}{k+1}$ so that the number of triangles with all three endpoints in small buckets (which may not be sampled) is sufficiently small (recall that the number of buckets is $k+1$). On the other hand, all other triangles have at least one of their three endpoints in a large bucket (which is sampled). This is the intuition for the source of the factor of $1/3$. For an illustration see Figure 1 in Subection 3.2.

3.1.2 The procedure Approx-Triangles-Degree

We next remove the assumption on having oracle access as described above by presenting a procedure named Approx-Triangles-Degree, which operates as follows (for an illustration see Figure 3 in Subsection 3.3.2). The procedure is invoked with a vertex v and an index j. Roughly speaking, the procedure determines whether v belongs to the bucket B_j. More precisely, if the vertex v belongs to a bucket $B_{j'}$ such that $j' \geq j-1$, then the procedure returns an estimation of the triangles-degree of the vertex up to a multiplicative factor of $(1 \pm \delta)$ for some small estimation error δ. Otherwise the procedure returns an estimate that is sufficiently smaller than $(1+\beta)^j$ (thus indicating that $v \notin B_j$). The procedure has expected query complexity and running time of roughly $O\left(\frac{d(v)\sqrt{m}}{(1+\beta)^j}\right)$, where $d(v)$ is the (neighbor) degree of the vertex v.

We use different methods to estimate the triangles-degree of a vertex v depending on its degree, $d(v)$. Specifically, we classify the graph vertices into two types: *low-degree vertices*, with $d(v) \leq \sqrt{m}$, and *high-degree vertices*, with $d(v) > \sqrt{m}$. To estimate the triangles-degree of a low-degree vertex v, we simply (repeatedly) sample uniformly a pair of its neighbors and check if there is an edge between the pair (in which case a triangle is observed). Denote by $\Delta(v)$ the triangles-degree of the vertex v. The probability of observing a triangle is $\Delta(v)/(d(v))^2$, which for a low-degree vertex v that belongs to bucket B_j is $\Omega\left(\frac{(1+\beta)^j}{d(v)\sqrt{m}}\right)$.

For high-degree vertices, the probability of observing a triangle using this procedure may be too low, implying that the number of queries necessary for estimating their triangles-degree in this manner may be too high. Therefore we use a different method to estimate their triangles-degree. Observe that a high-degree vertex can participate in two types of triangles: triangles in which all three endpoints are high-degree vertices, and triangles with at least one endpoint that is a low-degree vertex. We call the former type *high triangles*, and the latter *crossing triangles*. In order to estimate the number of high triangles that a high-degree vertex v participates in we modify the aforementioned sampling procedure so that it performs pair queries on uniformly selected pairs of high-degree neighbors of v.

In order to estimate the number of crossing triangles that a high-degree vertex v participates in we (roughly) do the following. We uniformly sample low-degree neighbors of v, and for each such selected neighbor u, we uniformly sample a neighbor w and check whether there is an edge between w and v (so that (v, u, w) is a triangle). We show that for both estimation procedures it is sufficient to perform a number of queries of roughly $O\left(\frac{d(v)\sqrt{m}}{(1+\beta)^j}\right)$. Observe that the sampling complexity grows linearly with the degree, and inversely with the triangles-degree of the vertex.

3.1.3 Significant buckets and useful tradeoffs

The dependence on j in the query complexity and running time of the Approx-Triangles-Degree procedure leads us to refine the threshold for largeness of buckets and we introduce the notion of "significance". Significant buckets are those with a significant contribution to the total number of triangles in the graph. We say that the j^{th} bucket B_j is *significant* if $|B_j| \geq \frac{\beta\overline{\Delta}}{(k+1)\cdot(1+\beta)^j}$ (recall that $\overline{\Delta}$ is within a constant factor from $\Delta(G)$ and that there are $k+1$ buckets). In order to hit such a bucket and estimate its size it suffices to take a sample whose size grows (roughly) like $\frac{n}{|B_j|}$, which is upper bounded by $O\left(\frac{n\cdot(1+\beta)^j\cdot(k+1)}{\Delta(G)}\right)$. This enables us to benefit from the following tradeoffs.

1. While the complexity of the procedure Approx-Triangles-Degree increases as j decreases, the size of the sample sufficient for estimating the size of B_j decreases as j decreases. Indeed, the product of the two does not depend on j. However, this product does depend on $d(v)$, which may be large.

2. For any degree d, the number of vertices with degree at least d is upper bounded by $\frac{2m}{d}$. If we take a sample of s vertices, then we do not expect to get many more than $\frac{s}{n} \cdot \frac{2m}{d}$ vertices with degree greater than d. So while the complexity of Approx-Triangles-Degree increases with the degree of the vertex it is given, the number of vertices with at least any given degree, decreases with the degree.

3.1.4 Buckets defined based on random thresholds

In order to benefit from the first aforementioned tradeoff we estimate the size of each bucket B_j using a separate sample whose size depends on j. This, together with the fact that the procedure Approx-Triangles-Degree only returns an estimate of $\Delta(v)$, gives rise to an additional difficulty. Namely, this could lead to an overestimation or underestimation of a bucket's size due to vertices that have a triangles-degree close to the bucket's boundaries ($(1+\beta)^{j-1}$ and $(1+\beta)^j$). To deal with this difficulty we redefine the buckets' boundaries in a random fashion so as to achieve two properties. The first property is that within each bucket all the vertices have the same triangles-degree up to a multiplicative factor of $(1\pm\beta)^2$. The second property is that the number of vertices with triangles-degree that is in a small range surrounding a bucket's boundary is small. We prove that these properties can be obtained (with high probability) when randomly choosing the boundaries between the buckets.

3.1.5 A $(1\pm\epsilon)$-approximation algorithm

In what follows, we say that a bucket B_j is large if $|B_j| \geq \max\left\{\frac{(\beta\cdot\overline{\Delta})^{1/3}}{k+1}, \frac{\beta\overline{\Delta}}{(k+1)\cdot(1+\beta)^j}\right\}$. As illustrated in Figure 1, we partition the graph's triangles into four subsets of triangles: $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}$, with all three endpoints in large buckets, $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}$, with two endpoints in large buckets, $\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}$, with one endpoint in a large bucket, and $\mathrm{Tr}_{\mathcal{S},\mathcal{S},\mathcal{S}}$, with no endpoint in a large bucket. The number of triangles in $\mathrm{Tr}_{\mathcal{S},\mathcal{S},\mathcal{S}}$ is sufficiently small, so that they can be disregarded (estimated as 0). Roughly speaking, the $\frac{1}{3}$-approximation algorithms sketched above obtains an estimate of $3|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}| + 2|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}| + |\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}|$ (and divides it by 3). The source of the different factors of 3, 2 and 1, is the following. The algorithm obtains a good estimate, $\hat{b}_j$, of the size of each large bucket B_j, and hence the sum over all large buckets B_j of $\hat{b}_j\cdot(1+\beta)^j$ accounts for 3 "copies" of each triangle in $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}$ (one for each endpoint), 2 "copies" of each triangle in $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}$ and one "copy" for each triangle in $\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}$.

The $(1\pm\epsilon)$-approximation algorithm aims at estimating $|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}|$ and $|\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}|$. If we could sample triangles uniformly among triangles that have an endpoint in a large bucket and could determine to which buckets the other endpoints belong, then we would obtain such estimates. The algorithm performs these tasks approximately, which is sufficient for our needs, and furthermore, does so without increasing the complexity of the algorithm by more than a $\mathrm{poly}(\log n, 1/\epsilon)$ factor.

Roughly speaking, the algorithm chooses a large bucket B_j with probability proportional to the number of triangles that have an endpoint in B_j. This is done using the approximated sizes of the large buckets and the approximated number of triangles with an endpoint in a large bucket, which were obtained by the $\frac{1}{3}$-approximation algorithm. Next, the algorithm samples a vertex v in B_j and uses a modified version of Approx-Triangles-Degree in order to sample a triangle incident to v. Finally, the algorithm determines for each of the two other endpoints of the sampled triangle whether it belongs to a large bucket or a small bucket.

Since we introduce quite a lot of notations, we gathered them in Table 1.

Table 1: Notations, their meaning, and their place of definition.

Notation	Meaning	Where defined
$\Delta(G)$, $\mathcal{T}(G)$	The number of unlabeled triangles in the graph G	Section 1.1
$\widehat{\Delta}$	The output of the algorithm – a $(1 \pm \epsilon)$ approximation of $\Delta(G)$	Section 1.1
ϵ	An approximation parameter	Section 1.1
$\Gamma(v)$, $d(v)$	The set of neighbors of v, and v's degree	Section 2
$\mathrm{Tr}(v)$, $\Delta(v)$	Number of triangles that v participates in, and v's triangles-degree	Section 2
$\overline{\Delta}$	An initial estimate of $\Delta(G)$, $\frac{\Delta(G)}{c_\Delta} \leq \overline{\Delta} \leq \Delta(G)$	Assumption 2.1
$\overline{m}$	An initial estimate of the number of edges m, $\overline{m} \geq \frac{m}{c_m}$	Assumption 2.1
$\mathcal{T}(G)$	Set of labeled triangles in the graph G	Section 3.2
$\overline{\mathcal{T}}$	$\overline{\mathcal{T}} = 3\overline{\Delta}$, $\frac{\lvert\mathcal{T}(G)\rvert}{c_\Delta} \leq \overline{\mathcal{T}} \leq \lvert\mathcal{T}(G)\rvert$	Section 3.2 and Equation (4)
$\mathcal{T}(A)$	Set of labeled triangles rooted at vertices in $A \subseteq V$	Section 3.2
β	$\epsilon/450$	Definition 3.2.1
$B_j = \widetilde{B}_j$	The j^{th} bucket before the random threshold process, $B_j = \{v : \Delta(v) \in ((1+\beta)^{j-1}, (1+\beta)^j]\}$	Definition 3.2.1
k	Number of buckets, $\log_{(1+\beta)} \overline{\mathcal{T}}$	Definition 3.2.1
I_j	Safety interval of the j^{th} bucket	Create-Random-Thresholds, Figure 2
μ_j	Midpoint of I_j and the new threshold	Create-Random-Thresholds, Figure 2
B_j	The j^{th} bucket, $B_j = \{v : \Delta(v) \in (\mu_{j-1}, \mu_j]\}$	Definition 3.2.1, Figure 2
B'_j	$B_j \setminus (B_{I_j} \cup B_{I_{j-1}})$, Strict buckets	Equation (9), Figure 2
B''_j	$B'_j \cup B_{I_j} \cup B_{I_{j-1}}$	Section 3.3.1
V_{hi}, $V_{\ell o}$	Set of high degree and low degree vertices, $V_{hi} = \{v : d(v) > \sqrt{m}\}$, $V_{lo} = \{v : d(v) \leq \sqrt{m}\}$	Subsection 3.3.2
$\Gamma_{hi}(v)$, $\Gamma_{\ell o}(v)$	Set of v's neighbors in V_{hi} and $V_{\ell o}$ respectively	Subsection 3.3.2
$\mathrm{Tr}_{hi}(v)$, $\Delta_{hi}(v)$	Set of high triangles v participates and their cardinality	Definition 3.3.3
$\mathrm{Tr}_{cr}(v)$, $\Delta_{cr}(v)$	Set of crossing triangles v participates and their cardinality	Definition 3.3.3
$\widehat{\Delta}(v)$	An estimate of the triangles degree of v	Approx-Triangles-Degree

$\mathcal{L}^*$	Set of indices of the large significant buckets, $\mathcal{L}^* = \left\{ j \in [k] : \ \lvert B'_j \rvert \geq \max\left\{ \frac{(\beta \cdot \overline{\mathcal{T}})^{1/3}}{k+1}, \frac{\beta \overline{\mathcal{T}}}{(k+1)\cdot \mu_j} \right\} \right\}$	Definition 3.2.2
$\mathcal{S}^*$	Set of indices of the small and insignificant buckets, $\mathcal{L}^* = [k] \setminus \mathcal{L}^*$	Definition 3.2.2
k'	Maximal triangles-degree of large significant buckets, $\log_{(1+\beta)} \frac{c_\Delta \cdot (k+1) \cdot \overline{\mathcal{T}}^{2/3}}{\beta}$	Definition 3.3.12
$\mathcal{L}$	Subset of indices $j \in [k']$ of the significant buckets, $\mathcal{L} = \left\{ j \in [k'] : \ \lvert B'_j \rvert \geq \frac{\beta \cdot \overline{\mathcal{T}}}{(k+1)\cdot \mu_j} \right\}$	Definition 3.3.12
B'_X	$\bigcup_{j \in X} B'_j$	Definition 3.3.7
$\mathrm{Tr}_{X,Y,Z}$	Triangles with endpoints in B'_X, B'_Y, B'_Z	Definition 3.3.8
$\widehat{b}_j$	An estimate for $\lvert B'_j \rvert$	$\frac{1}{3}$-Approx-Triangles
$\widehat{\mathcal{L}}$	$\widehat{\mathcal{L}} = \{ j : \ \widehat{b}_j \geq (1-\beta) \frac{\beta \cdot \overline{\mathcal{T}}}{k' \cdot \mu_j} \}$, $\widehat{\mathcal{L}} \supseteq \mathcal{L} \supseteq \mathcal{L}^*$	$\frac{1}{3}$-Approx-Triangles
$\widehat{\mathcal{S}}$	Set of indices of the small significant buckets, $\widehat{\mathcal{S}} = \left\{ j \in [k] \setminus \widehat{\mathcal{L}} \ : \ B'_j \geq \frac{\beta \overline{\mathcal{T}}}{(k+1)\cdot \mu_j} \right\}$	Definition 3.4.1
$\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$	$\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = \lvert \mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \rvert / \lvert \mathcal{T}(B'_L) \rvert$	Definition 3.4.2
$\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$	$\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} = \lvert \mathrm{Tr}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \rvert / \lvert \mathcal{T}(B'_L) \rvert$	Definition 3.4.2
q	The "real" query complexity, $q = \max\left\{ \frac{n}{\Delta^{1/3}}, \min\left\{ m, \frac{m^{3/2}}{\Delta(G)} \right\} \right\}$	Discussion in Subsection 3.5.2
$\overline{q}$	The "guessed" value of q	Approx-Triangles
$\overline{\Delta}_{\overline{q},i}$	$\overline{\Delta}_{\overline{q},i} = \frac{n^3}{\overline{q} \cdot 2^i}$	Approx-Triangles
$\overline{m}_{\overline{q},i}$	$\overline{m}_{\overline{q},i} = \max\left\{ \overline{q}, \frac{n^2}{2^{2i/3}} \right\}$	Approx-Triangles

3.2 A 1/3-approximation given oracle access to $\Delta(v)$

In what follows we think of each triangle (u, v, w) as having three copies, each labeled with one of its endpoints: $(u, v, w)_u, (u, v, w)_v$ and $(u, v, w)_w$. We say that a labeled copy of a triangle $(u, v, w)_u$ is a labeled triangle that is rooted at u. Let $\mathcal{T}(G)$ denote the set of labeled (rooted) triangles. Clearly,

$$\Delta(G) = \frac{1}{3} |\mathcal{T}(G)|. \tag{3}$$

Therefore, to get an approximation of $\Delta(G)$ it suffices to approximate $|\mathcal{T}(G)|$. We define $\overline{\mathcal{T}} = 3\overline{\Delta}$, and it follows from the first item in Assumption 2.1 and Equation (3) that

$$\frac{|\mathcal{T}(G)|}{c_\Delta} \leq \overline{\mathcal{T}} \leq |\mathcal{T}(G)|. \tag{4}$$

In general, for a set of vertices $A \subseteq V$, we denote by $\mathcal{T}(A)$ the set of all labeled triangles rooted at vertices in A.

Definition 3.2.1 *Let $\beta = \epsilon/450$ and let $k = \log_{(1+\beta)} \overline{\mathcal{T}}$ (so that $k = O(\log n/\epsilon)$). For $j = 0, \ldots, k$, let the bucket B_j be:*

$$B_j = \{v : \Delta(v) \in ((1+\beta)^{j-1}, (1+\beta)^j]\}. \tag{5}$$

In what follows we denote by $[k]$ the set $\{0, \ldots, k\}$ If for each $j \in [k]$ we could get an estimate $\widehat{b}_j$ for $|B_j|$ such that $(1-\beta)|B_j| \leq \widehat{b}_j \leq (1+\beta)|B_j|$, then by defining $\widehat{t}$:

$$\widehat{t} = \sum_{j \in [k]} \widehat{b}_j \cdot (1+\beta)^j, \tag{6}$$

we would get

$$(1-\beta)|\mathcal{T}(G)| \leq \widehat{t} \leq (1+\beta)^2|\mathcal{T}(G)|. \tag{7}$$

The problem is that in order to estimate the size of a bucket B_j by sampling, the sample size should be at least $\Omega\left(\frac{n}{|B_j|}\right)$. For "small" buckets with respect to n this gives a high dependence on n. Therefore we will only estimate the sizes of "large" buckets for some appropriate threshold of "largeness". Using the estimated sizes of the large buckets, we can obtain an estimate of the number of triangles that have at least one endpoint in a large bucket. We will show that this gives an approximation $\widehat{t}$ of $|\mathcal{T}(G)|$ such that $\frac{1}{3}(1-O(\beta))|\mathcal{T}(G)| \leq \widehat{t} \leq (1+O(\beta))|\mathcal{T}(G)|$.
We start with introducing the following definitions:

Definition 3.2.2 *We say that a bucket B_j is* large *if $|B_j| \geq \frac{(\beta\overline{\mathcal{T}})^{1/3}}{k+1}$ and is* small *otherwise. We denote by $\mathcal{L}^*$ the set of indices of the large buckets, and by $\mathcal{S}^*$ the set of indices of small buckets. Namely, $\mathcal{L}^* = \left\{j \in [k] : |B_j| \geq \frac{(\beta\overline{\mathcal{T}})^{1/3}}{k+1}\right\}$ and $\mathcal{S}^* = [k] \setminus \mathcal{L}^*$.*

Definition 3.2.3 *For a set of indices $X \subseteq [k]$, let B_X denote the union of all the buckets B_j for which $j \in X$. That is, $B_X = \bigcup_{j \in X} B_j$.*

Definition 3.2.4 *For sets on indices $X, Y, Z \subseteq [k]$, let $Tr_{X,Y,Z}$ denote the set of triangles (u, v, w) such that $u \in B_X$, $v \in B_Y$ and $w \in B_W$.*

Claim 3.2.1 *There are at most $\beta|\mathcal{T}(G)|$ labeled triangles with all three endpoints in small buckets.*

Proof: Since there are $k+1$ buckets, there are at most $(\beta\overline{\mathcal{T}})^{1/3}$ vertices in the small buckets, which form at most $\beta\overline{\mathcal{T}}$ labeled triangles with one another. By Equation (4), $\overline{\mathcal{T}} \leq |\mathcal{T}(G)|$, and the claim follows. ■

Claim 3.2.2 *For $\mathcal{L}^*$ as defined in Definition 3.2.2,*

$$\frac{1}{3}(1-9\beta)|\mathcal{T}(G)| \leq |\mathcal{T}(B_{\mathcal{L}^*})| \leq |\mathcal{T}(G)|.$$

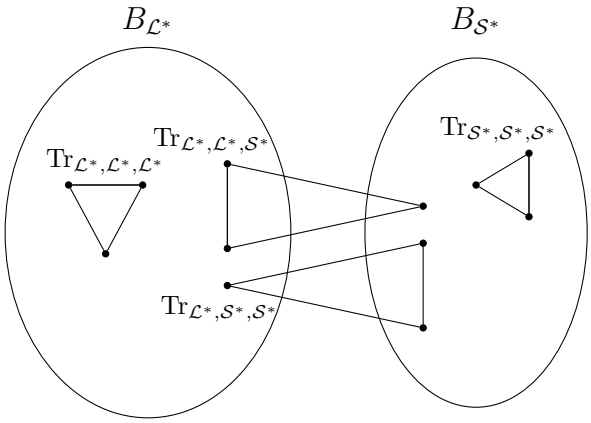


Figure 1: An illustration of the different types of triangles referred to in Claim 3.2.2

.

Proof: Clearly,

$$|\mathcal{T}(B_{\mathcal{L}^*})| \leq |\mathcal{T}(G)|.$$

To prove the lower bound on $|\mathcal{T}(B_{\mathcal{L}^*})|$, consider the following sets of triangles: $Tr_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{L}^*}$, $Tr_{\mathcal{L},\mathcal{L},\mathcal{S}}$, $Tr_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}$ and $Tr_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}$. Observe that these sets partition the graph's triangles into four disjoint sets (see Figure 1 for an illustration). Therefore, it holds that

$$|\mathcal{T}(G)| = 3|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{L}^*}| + 3|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| + 3|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| + 3|\mathrm{Tr}_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}|. \tag{8}$$

By the definition of $\mathcal{T}(B_{\mathcal{L}^*})$,

$$\begin{aligned}|\mathcal{T}(B_{\mathcal{L}^*})| &= 3|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{L}^*}| + 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| + |\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| \\ &= |\mathcal{T}(G)| - 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| - |\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| - 3|\mathrm{Tr}_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}|\ .\end{aligned}$$

Also, by Equation (8),

$$2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| + |\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| \leq 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| + 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| \leq \frac{2}{3}|\mathcal{T}(G)|.$$

Observe that $\mathrm{Tr}_{\mathcal{S},\mathcal{S},\mathcal{S}}$ is exactly the set described in Claim 3.2.1. Therefore,

$$3|\mathrm{Tr}_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}| \leq 3\beta|\mathcal{T}(G)|.$$

Hence,

$$|\mathcal{T}(B_{\mathcal{L}^*})| \geq \frac{1}{3}|\mathcal{T}(G)| - 3\beta|\mathcal{T}(G)| = \frac{1}{3}(1-9\beta)|\mathcal{T}(G)|,$$

and the proof is complete. ■

Assume we had access to a triangles-degree oracle $\mathcal{O}_\Delta$ such that when queried on a vertex v would return $\Delta(v)$. If we set the threshold for largeness at $\frac{(\beta\cdot\overline{\mathcal{T}})^{1/3}}{k+1}$ (as in Definition 3.2.2), a sample of size $\Theta\left(\frac{n}{\overline{\mathcal{T}}^{1/3}}\right)\cdot \mathrm{poly}(\log n, 1/\epsilon)$ would suffice to estimate the sizes of the large buckets and obtain an approximation $\widehat{t}$ of $|\mathcal{T}(G)|$ such that $\frac{1}{3}(1-O(\beta))|\mathcal{T}(G)| \leq \widehat{t} \leq (1+O(\beta))|\mathcal{T}(G)|$.

3.3 A 1/3-approximation algorithm

In this subsection we remove the assumption on having oracle access to the triangles-degree of the vertices. We describe a $\frac{1}{3}$-approximation algorithm for the number of triangles in a graph using only queries to the graph (as defined in the preliminaries). We start by providing several building blocks that will be used in the design of the algorithm.

3.3.1 Random threshold partitioning

As we show in the following subsections, in order to obtain the complexity we desire, we need to estimate the size of each bucket j using a separate sample whose size depends on j. This, together with the fact that for every vertex, we will only have an estimate of its triangles-degree, may lead to an overestimation or underestimation of a bucket's size, $|B_j|$, due to vertices that have a triangles-degree close to the bucket's boundaries ($(1+\beta)^{j-1}$ and $(1+\beta)^j$). The reason is that it is possible that, when estimating $|B_j|$, vertices that belong to B_{j-1} but whose triangles-degree is very close to the boundary with B_j (i.e., $(1+\beta)^{j-1}$), will be assigned to B_j, and when estimating $|B_{j-1}|$ such sampled vertices will be assigned to B_{j-1}. If there are many such vertices, then this may result in an overestimation. Alternatively, it is possible that such vertices will not be assigned to either bucket when the size of the corresponding bucket is estimated, resulting in an underestimation.

To deal with this difficulty we redefine the buckets' boundaries in a random fashion so as to achieve two properties. The first property is that within each bucket all the vertices have the same triangles-degree up to a multiplicative factor of $(1 \pm \beta)^2$. The second property is that the number of vertices with triangles-degree that is in a small range surrounding a bucket's boundary is small. We prove that these properties can be obtained with high constant probability when randomly choosing the boundaries between the buckets.

We use the following procedure to set the buckets' boundaries.

Procedure 1 Create-Random-Thresholds (β, γ, k)

1: **For** $j = 0, \ldots, k-1$ **do**
2: Partition the range $((1+\beta)^{j-1}, (1+\beta)^j]$ into $\frac{\log n}{\gamma}$ intervals of equal size $I_j^1, \ldots, I_j^{\frac{\log n}{\gamma}}$.
3: Choose uniformly at random one of the intervals.
4: Denote the selected interval the "safety interval" I_j, and its midpoint μ_j.
5: Let $\mu_{-1} \leftarrow (1+\beta)^{-1}$ and $\mu_k \leftarrow (1+\beta)^k$.
6: Return $\{\mu_{-1}, \ldots, \mu_k\}$.

We start by introducing some notations.

- Denote by $\widetilde{B}_j$ the buckets defined in Definition 5.
- For $j \in [k]$ let the (new) j^{th} bucket be

$$B_j = \{v : \Delta(v) \in (\mu_{j-1}, \mu_j]\} ,$$

where the μ_j values are as defined in Step (4) of Create-Random-Thresholds.

- For every $1 \leq \ell \leq \frac{\log n}{\gamma}$ denote by $B_{I_j^\ell}$ the set of vertices with triangles-degree in I_j^ℓ. Namely,

$$B_{I_j^\ell} = \left\{ v : \ \Delta(v) \in I_j^\ell \right\},$$

  where the I_j^ℓ intervals are as defined in Step (2) of Create-Random-Thresholds procedure.

- Denote by B'_j the subset of vertices that belong to the bucket B_j, whose triangles-degree is not in a safety interval. Namely,

$$B'_j = B_j \setminus (B_{I_j} \cup B_{I_{j-1}}). \tag{9}$$

  We refer to the subsets B'_j as strict buckets.

- Let $B''_j = B'_j \cup B_{I_j} \cup B_{I_{j-1}}$.

See Figure 2 for an illustration of the above notations.

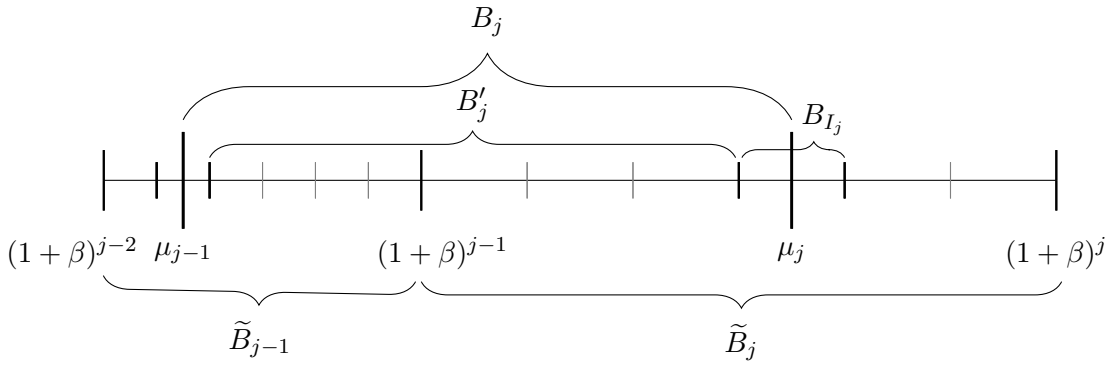


Figure 2: An illustration of the new buckets ranges.

Definition 3.3.1 *We say that the selection of the safety intervals in Create-Random-Thresholds is a good selection if for every $j \in [k]$, it holds that $|B_{I_j}| \leq \beta |\widetilde{B}_j|$.*

Corollary 3.3.1 *If the selection of the safety intervals is good, then for every $j \in [k]$,*

$$|B''_j| = |B'_j \cup B_{I_j} \cup B_{I_{j-1}}| \leq |B'_j| + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|.$$

Claim 3.3.2 *Procedure Create-Random-Thresholds outputs a good selection with probability at least $1 - \frac{1}{4\log^3 n}$.*

Proof: Fix a choice of $j \in [k]$. There can be at most $\frac{1}{\beta}$ intervals such that $|B_{I_j^\ell}| > \beta|\widetilde{B}_j|$. We refer to these intervals as "heavy intervals". Since we partition each range into $\frac{\log n}{\gamma}$ intervals, the probability of randomly selecting a heavy interval is at most $\frac{\gamma}{\beta \log n}$. By applying the union bound over all j's we get that with probability at least $1 - \frac{\gamma}{\beta}$, for every j in $[k]$,

$$|B_{I_j}| \leq \beta|\widetilde{B}_j|.$$

Setting $\gamma = \frac{\beta}{4\log^3 n}$ completes the proof. ■

Claim 3.3.3 *If the selection of the safety intervals in* Create-Random-Thresholds *is a good selection, then*

$$(1-3\beta)|\mathcal{T}(G)| \leq \sum_{j\in[k]} |B'_j| \cdot \mu_j \leq (1+\beta)^2 \cdot |\mathcal{T}(G)|.$$

Proof: By the setting of the μ_j values in Create-Random-Thresholds, for every $j \in [k]$ it holds that $\mu_{j-1} \leq \mu_j \leq (1+\beta)^2 \cdot \mu_{j-1}$. Also, for every $v \in B_j$ it holds that $\mu_{j-1} \leq \Delta(v) \leq \mu_j$. Therefore, $\mu_j \leq (1+\beta)^2\Delta(v)$ and

$$\sum_{j\in[k]} |B'_j| \cdot \mu_j \leq \sum_{j\in[k]} |B_j| \cdot \mu_j \leq (1+\beta)^2 \cdot |\mathcal{T}(G)|.$$

From the definition of B'_j and of a good selection,

$$\begin{aligned}\sum_{j\in[k]} |B'_j| \cdot \mu_j &\geq \sum_{j\in[k]} (|B_j| - \beta|\widetilde{B}_j| - \beta|\widetilde{B}_{j-1}|) \cdot \mu_j \\ &= \sum_{j\in[k]} |B_j| \cdot \mu_j - \sum_{j\in[k]} \beta|\widetilde{B}_j| \cdot \mu_j - \sum_{j\in[k]} \beta|\widetilde{B}_{j-1}| \cdot \mu_j \ .\end{aligned}$$

By the selection of μ_j, for every j it holds that $\mu_j \leq (1+\beta)^j$. Hence,

$$\begin{aligned}\sum_{j\in[k]} |B'_j| \cdot \mu_j &\geq \sum_{j\in[k]} |B_j| \cdot \mu_j - \sum_{j\in[k]} \beta|\widetilde{B}_j| \cdot (1+\beta)^j - \sum_{j\in[k]} \beta|\widetilde{B}_{j-1}| \cdot (1+\beta)(1+\beta)^{j-1} \\ &\geq |\mathcal{T}(G)| - \beta|\mathcal{T}(G)| - (1+\beta)\beta|\mathcal{T}(G)| \\ &\geq (1-3\beta)|\mathcal{T}(G)|,\end{aligned}$$

and the proof is complete. ■

Therefore in what follows we will aim to approximate the sizes of the strict buckets.

3.3.2 Approximating the triangles-degree of a vertex

In this subsection we present a procedure for estimating the triangles-degree of a vertex. To this end, we partition the graph vertices into two disjoint sets – high-degree vertices denoted by V_{hi} and low-degree vertices, denoted by $V_{\ell o}$. Namely,

$$V_{hi} = \{v \mid d(v) > \sqrt{\overline{m}}\} \text{ and } V_{\ell o} = \{v \mid d(v) \leq \sqrt{\overline{m}}\}.$$

Additional notations: For a vertex v we denote by $\Gamma_{hi}(v)$ the set of v's neighbors that belong to V_{hi}, and by $\Gamma_{\ell o}(v)$ the set of v's neighbors that belong to $V_{\ell o}$.

Observation 3.1 *By the definition of V_{hi}, and by the assumption that $\frac{m}{c_m} \leq \overline{m}$, we have that $|V_{hi}| < \frac{2m}{\sqrt{\overline{m}}} \leq \frac{2c_m \cdot \overline{m}}{\sqrt{\overline{m}}} = O(\sqrt{\overline{m}})$.*

We also partition all the triangles in the graph into three disjoint sets:

- Triangles with all three endpoints in V_{hi}, referred to as High triangles.

- Triangles with endpoints both in $V_{\ell o}$ and in V_{hi}, referred to as Crossing triangles.
- Triangles with all three endpoints in $V_{\ell o}$.

We use different methods to approximate the triangles-degree of high-degree vertices and of low-degree vertices.

- Given a low-degree vertex v (i.e., for which $d(v) \leq \sqrt{\overline{m}}$), we uniformly sample pairs of neighbors, $u, w \in \Gamma(v)$, and for each sampled pair we make a pair query to determine whether $(u, w) \in E$. If $\Delta(v) = \Omega(\mu_j)$, then the probability of hitting a triangle (v, u, w) is $\Omega\left(\frac{\mu_j}{d(v)^2}\right)$. Therefore, if we sample $\Theta\left(\frac{d(v)^2}{\mu_j} \cdot \frac{\log n}{\delta^2}\right)$ pairs of neighbors of v, then we can obtain an estimate $\widehat{\Delta}(v)$ that with probability $1 - n^{-2}$ approximates $\Delta(v)$ to within a factor of $(1 \pm \delta)$. Since $d(v) \leq \sqrt{\overline{m}}$, the number of queries performed is $O\left(\frac{d(v) \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \frac{\log n}{\delta^2}\right)$, as desired.
- Consider now a high-degree vertex v, that is, for which $d(v) > \sqrt{\overline{m}}$. For such a vertex, the query complexity of the procedure just described for low-degree vertices may be too high. Therefore, we use a different procedure, which in particular separately estimates the number of high triangles rooted at v and the number of crossing triangles rooted at v. In what follows we assume that we know $|\Gamma_{hi}(v)|$ (where in reality, we shall use an estimate of this size).
  - Consider first high triangles rooted at v, and assume that the number of such triangles is $\Omega(\delta \cdot \mu_j)$ (or else their contribution to $\Delta(v)$ is negligible). Suppose we could uniformly sample pairs of neighbors of v that belong to V_{hi}. Then, similarly to the discussion regarding low-degree vertices, a sample of $\Theta\left(\frac{|\Gamma_{hi}(v)|^2}{\delta \cdot \mu_j} \cdot \frac{\log n}{\delta^2}\right)$ pairs suffices to estimate the number of these triangles. Since $|V_{hi}| \leq 2\sqrt{\overline{m}}$ and $|\Gamma_{hi}(v)| \leq d(v)$, the number of queries is $O\left(\frac{d(v) \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \frac{\log n}{\delta^3}\right)$.

    In order to obtain such a sample of pairs of vertices in $\Gamma_{hi}(v)$, we simply take a sample of vertices from $\Gamma(v)$ whose size is $\Theta\left(\frac{d(v)}{|\Gamma_{hi}(v)|}\right)$ times larger, and perform a degree query on each sampled vertex so as to determine whether it belongs to V_{hi}. Using the fact that $|\Gamma_{hi}(v)| \leq |V_{hi}| \leq O(\sqrt{\overline{m}})$ the number of degree queries performed is as desired.
  - Consider next crossing triangles rooted at v. By the definition of crossing triangles, for each such triangle (v, u, u'), either $u \in V_{\ell o}$ or $u' \in V_{\ell o}$ (or both). Assume without loss of generality that $u \in V_{\ell o}$. In this case we would like to exploit the fact that $d(u) \leq \sqrt{\overline{m}}$. Consider all triples (v, u, ℓ) where $u \in \Gamma(v)$ and $1 \leq \ell \leq \sqrt{\overline{m}}$. The number of such triples is $d(v) \cdot \sqrt{\overline{m}}$. Such a triple corresponds to a crossing triangle rooted at v if $u \in \Gamma_{\ell o}(v)$ and the ℓ^{th} neighbor of u is also a neighbor of v. This can be verified by performing one degree query, one neighbor query, and one pair query.

    Note that triangles $(v, u, u')_v$ such that both u and u' are in $V_{\ell o}$ are twice as likely to be sampled compared to triangles (v, u, u') such that either u or u', but not both, are in $V_{\ell o}$. Therefore when we hit a triangle of the former type we consider it as "half a triangle". If the number of crossing triangles rooted at v is $\Omega(\delta \cdot \mu_j)$, then the number of queries sufficient for approximating $\Delta(v)$ to within $(1 \pm \delta)$ is as desired.

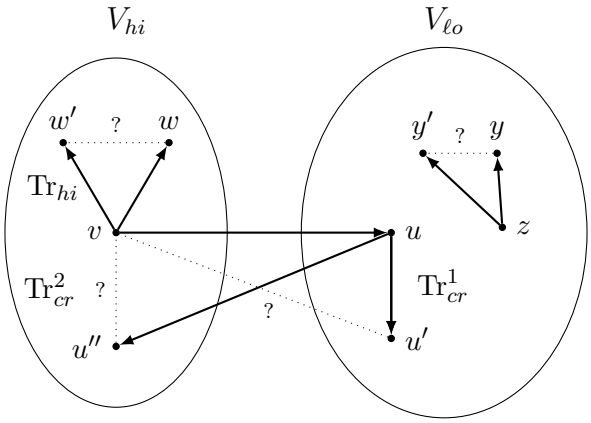


Figure 3: An illustration for the procedure Approx-Triangles-Degree. $V_{\ell o}$ denotes the set of low-degree vertices, and V_{hi} denotes the set of high-degree vertices. For a vertex $v \in V_{hi}$, there are three types of triangles that v can be an endpoint of: Triangles in which the two other endpoints also belong to V_{hi} (Tr_{hi}), triangles in which the two other endpoints are in $V_{\ell o}$ (Tr^1_{cr}) and triangles in which one endpoint is in $V_{\ell o}$ and the other in V_{hi} (Tr^2_{cr}).

We first present the three aforementioned sub-procedures, starting with the procedure for approximating the triangles-degree of a low-degree vertex.

Algorithm 2 Approx-Triangles-Of-Low-Deg-Vertices$(v, j, d(v), \delta)$

1: $r \leftarrow 0$.
2: **Repeat** $s = \frac{d(v)^2}{\delta \cdot \mu_{j-1}} \cdot \frac{20 \log n}{\delta^2}$ times:
3: Uniformly independently at random choose $u, u' \in \Gamma(v)$.
4: **If** $(u, u') \in E$ **then**
5: $r \leftarrow r + 1$.
6: $\widehat{\Delta} \leftarrow r \cdot \frac{d(v)^2}{s}$.
7: **Return** $\widehat{\Delta}$.

Definition 3.3.2 *We say that Approx-Triangles-Of-Low-Deg-Vertices$(v, j, d(v), \delta)$ answers correctly if the procedure returns $\widehat{\Delta}$ for which the following holds. If $\Delta(v) \geq \delta \cdot \mu_{j-1}$, then $(1-\delta)\Delta(v) \leq \widehat{\Delta} \leq (1+\delta)\Delta(v)$ and otherwise $\widehat{\Delta} \leq (1+\delta) \cdot \delta \cdot \mu_{j-1}$.*

Claim 3.3.4 *For every v and for every $j \in [k']$, the procedure Approx-Triangles-Of-Low-Deg-Vertices$(v, j, d(v), \delta)$ answers correctly, as defined in Definition 3.3.2, with probability at least $1 - \frac{1}{n^3}$.*

Proof: Let $\chi_1, \ldots, \chi_s$ be Bernoulli random variables such that $\chi_i = 1$ if a sampled pair u, u' from Step sample two random neighbors in Approx-Triangles-Of-Low-Deg-Vertices is such that $(v, u, u') \in \Delta(G)$. It holds that $\mathbb{E}[\chi_i] = \frac{\Delta(v)}{d(v)^2}$ and $r = \sum_{i=1}^{s} \chi_i$. Applying the multiplicative Chernoff bound we

get that if $\Delta(v) \geq \delta \cdot \mu_{j-1}$ then

$$\Pr\left[\frac{1}{s}r > (1+\delta)\frac{\Delta(v)}{d(v)^2}\right] < \exp\left(-\frac{\delta^2}{3}\cdot\frac{\Delta(v)}{d(v)^2}\cdot s\right) < \exp\left(-\frac{\delta^2}{3}\cdot\frac{\Delta(v)}{d(v)^2}\cdot\frac{d(v)^2}{\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2}\right) < \frac{1}{2n^3}.$$

Similarly $\Pr\left[\frac{1}{s}r < (1-\delta)\frac{\Delta(v)}{d(v)^2}\right] < \frac{1}{2n^3}$.
If $\Delta(v) < \delta \cdot \mu_{j-1}$ then

$$\Pr\left[\frac{1}{s}r > (1+\delta)\frac{\delta\cdot\mu_{j-1}}{d(v)^2}\right] < \exp\left(-\frac{\delta^2}{3}\cdot\frac{\delta\cdot\mu_{j-1}}{d(v)^2}\cdot s\right) < \frac{1}{n^3}.$$

Recalling that $\widehat{\Delta} = r \cdot \frac{d(v)^2}{s}$, the claim follows. ■

Claim 3.3.5 *For every vertex v and every index $j \in [k']$, the query complexity and running time of the procedure* Approx-Triangles-Of-Low-Deg-Vertices$(v, j, d(v), \delta)$ *are* $\frac{d(v)\cdot\sqrt{\overline{m}}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon)$.

Proof: In each iteration of the loop in Step (2), two neighbor queries are performed. All the other steps of the procedure require constant time and no queries. Therefore, the query complexity and running time of the procedure are $3s = 3 \cdot \frac{d(v)^2}{\delta\mu_{j-1}} \cdot \frac{20\log n}{\delta^2}$. Since $\mu_{j-1} \geq \frac{\mu_j}{(1+\beta)^2}$, $d(v) \leq \sqrt{\overline{m}}$ and $\delta = \mathrm{poly}(1/\epsilon)$ the claim follows. ■

Definition 3.3.3 *For a high-degree vertex v we denote by $Tr_{hi}(v)$ and $Tr_{cr}(v)$ its sets of high-triangles and crossing-triangles, respectively. We let $\Delta_{hi}(v) = |Tr_{hi}(v)|$ and $\Delta_{cr}(v) = |Tr_{cr}(v)|$.*

Algorithm 3 Approx-High-Triangles$(v, j, d(v), \delta)$

1: Uniformly independently at random sample $s = \frac{d(v)}{\sqrt{\delta\cdot\mu_{j-1}}} \cdot \frac{20\log n}{\delta^2}$ vertices from $\Gamma(v)$.
2: Denote the selected (multi) set S.
3: Query the degree of every vertex $u \in S$.
4: Let $\widehat{\Gamma}_{hi} \leftarrow |S \cap \Gamma_{hi}(v)| \cdot \frac{d(v)}{s}$.
5: **If** $\widehat{\Gamma}_{hi} < (1-\delta)\sqrt{\delta\cdot\mu_{j-1}}$ **then**
6: **Return** 0
7: $r_{hi} \leftarrow 0$.
8: Uniformly independently at random sample $s' = \frac{\widehat{\Gamma}_{hi}\cdot d(v)}{(1-\delta)\cdot\delta\cdot\mu_{j-1}} \cdot \frac{400\log n}{\delta^2}$ vertices from $\Gamma(v)$.
9: Denote the selected (multi) set S'.
10: Query the degree of every vertex $u \in S'$.
11: Let $S'_{hi} \leftarrow S' \cap \Gamma_{hi}(v)$, and $s'_{hi} \leftarrow \frac{1}{2}|S'_{hi}|$.
12: Partition S'_{hi} into pairs $S'_{hi} \leftarrow \{\{u_1, u'_1\}, \ldots \{u_{s'_{hi}}, u'_{s'_{hi}}\}\}$.
13: **For** every pair $\{u_i, u'_i\} \in S'_{hi}$ **do**
14: **If** $(u_i, u'_i) \in E$ **then**
15: $r_{hi} \leftarrow r_{hi} + 1$.
16: $\widehat{\Delta}_{hi} \leftarrow r_{hi} \cdot \frac{\widehat{\Gamma}^2_{hi}}{s'_{hi}}$
17: **Return** $\widehat{\Delta}_{hi}$.

Definition 3.3.4 *We say that* Approx-High-Triangles$(v, j, d(v), \delta)$ *answers correctly if the procedure returns* $\widehat{\Delta}_{hi}$ *for which the following holds. If* $\Delta_{hi}(v) \geq \delta \cdot \mu_{j-1}$*, then* $(1-\delta)\Delta_{hi}(v) \leq \widehat{\Delta}_{hi} \leq (1+\delta)\Delta_{hi}(v)$ *and otherwise* $\widehat{\Delta}_{hi} \leq (1+\delta) \cdot \delta \cdot \mu_{j-1}$.

Claim 3.3.6 *For every* v *and for every* $j \in [k']$*, with probability at least* $1 - \frac{1}{n^3}$*, the procedure* Approx-High-Triangles$(v, j, d(v), \delta)$ *answers correctly, as defined in Definition 3.3.4.*

Proof: First assume that $\Delta_{hi}(v) \geq \delta \cdot \mu_{j-1}$. Since $\Delta_{hi}(v) \leq |\Gamma_{hi}(v)|^2$, it holds that

$$|\Gamma_{hi}(v)| \geq \sqrt{\delta \cdot \mu_{j-1}}.$$

By the choice of s in Step (1),

$$\frac{|\Gamma_{hi}(v)|}{d(v)} \cdot s \geq \frac{\sqrt{\delta \cdot \mu_{j-1}}}{d(v)} \cdot \frac{d(v)}{\sqrt{\delta \cdot \mu_{j-1}}} \cdot \frac{20 \log n}{\delta^2} = \frac{20 \log n}{\delta^2}. \tag{10}$$

The probability of a sampled vertex in S to be in $\Gamma_{hi}(v)$ is $\frac{|\Gamma_{hi}(v)|}{d(v)}$. Hence, by applying the multiplicative Chernoff bound and by Equation (10), we have that:

$$\Pr\left[\frac{1}{s}|S \cap \Gamma_{hi}| > (1+\delta) \cdot \frac{|\Gamma_{hi}(v)|}{d(v)}\right] < \exp\left(-\frac{\delta^2}{3} \cdot \frac{|\Gamma_{hi}(v)|}{d(v)} \cdot s\right) < \frac{1}{8n^3}, \tag{11}$$

and

$$\Pr\left[\frac{1}{s}|S \cap \Gamma_{hi}| < (1-\delta) \cdot \frac{|\Gamma_{hi}(v)|}{d(v)}\right] < \exp\left(-\frac{\delta^2}{2} \cdot \frac{|\Gamma_{hi}(v)|}{d(v)} \cdot s\right) < \frac{1}{8n^3}. \tag{12}$$

By the setting of $\widehat{\Gamma}_{hi}$ to be $\widehat{\Gamma}_{hi} = |S \cap \Gamma_{hi}(v)| \cdot \frac{d(v)}{s}$, and by Equations (11) and (12), we get that with probability at least $1 - \frac{1}{4n^3}$,

$$(1-\delta) \cdot |\Gamma_{hi}(v)| \leq \widehat{\Gamma}_{hi} \leq (1+\delta) \cdot |\Gamma_{hi}(v)|. \tag{13}$$

Therefore, if $|\Gamma_{hi}(v)| > \sqrt{\delta \cdot \mu_{j-1}}$ then with probability at least $1 - \frac{1}{4n^3}$ we continue to approximate $\widehat{\Delta}_{hi}$.

In order to approximate $\widehat{\Delta}_{hi}$ correctly, S_{hi} should be sufficiently large. Namely, we need

$$|S'_{hi}| \geq \frac{|\Gamma_{hi}(v)|^2}{\delta \cdot \mu_{j-1}} \cdot \frac{20 \log n}{\delta^2}. \tag{14}$$

Denote the above quantity by s_d. We prove that $|S'_{hi}| \geq s_d$ with probability at least $1 - \frac{1}{4n^3}$. Let $\chi_1, \ldots, \chi_{s'}$ be Bernoulli random variables such that $\chi_i = 1$ if the i^{th} sampled vertex in S is in $\Gamma_{hi}(v)$, and 0 otherwise. It holds that $\mathbb{E}[\chi_i] = \frac{|\Gamma_{hi}(v)|}{d(v)}$ and that $|S_{hi}| = \sum_{i=1}^{s'} \chi_i$. Observe that by Equation (13), with probability at least $1 - \frac{1}{4n^3}$, it holds that $|\Gamma_{hi}(v)| \leq \widehat{\Gamma}_{hi}/(1-\delta)$. Therefore, by applying the multiplicative Chernoff and since $\widehat{\Gamma}_{hi} > (1-\delta) \cdot \sqrt{\delta \cdot \mu_{j-1}}$, we have that with

probability at least $1-\frac{1}{4n^3}$,

$$\begin{aligned}\Pr\left[\frac{1}{s'}|S'\cap\Gamma_{hi}(v)|<0.1\cdot\frac{|\Gamma_{hi}(v)|}{d(v)}\right] &< \exp\left(-\frac{0.9^2}{3}\cdot\frac{|\Gamma_{hi}(v)|}{d(v)}\cdot s'\right)\\ &= \exp\left(-\frac{0.9^2}{3}\cdot\frac{|\Gamma_{hi}(v)|}{d(v)}\cdot\frac{\widehat{\Gamma}_{hi}\cdot d(v)}{(1-\delta)\cdot\delta\cdot\mu_{j-1}}\cdot\frac{400\log n}{\delta^2}\right)\\ &= \exp\left(-\frac{0.9^2}{3}\cdot\frac{\widehat{\Gamma}^2_{hi}}{(1-\delta)^2\cdot\delta\cdot\mu_{j-1}}\cdot\frac{400\log n}{\delta^2}\right)<\frac{1}{4n^3}.\end{aligned}$$

Therefore with probability at least $1-\frac{1}{2n^3}$,

$$s'_{hi}=\frac{1}{2}|S'\cap\Gamma_{hi}(v)|\geq 0.05\cdot\frac{|\Gamma_{hi}(v)|}{d(v)}\cdot s'\geq\frac{\widehat{\Gamma}^2_{hi}}{(1-\delta)^2\cdot\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2}\geq\frac{|\Gamma_{hi}(v)|^2}{\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2},$$

and Equation (14) hold. If Equation (14) holds, then by the assumption that $\Delta_{hi}(v)\geq\delta\cdot\mu_{j-1}$, we have that:

$$\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}\cdot s'_{hi}\geq\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}\cdot\frac{\widehat{\Gamma}_{hi}(v)^2}{(1-\delta)^2\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2}\geq\frac{\Delta_{hi}(v)}{\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2}\geq\frac{20\log n}{\delta^2}. \tag{15}$$

The probability of a pair of sampled vertices $u,u'\in\Gamma_{hi}(v)$ to form a triangle with v is $\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}$. Therefore, by the multiplicative Chernoff bound and by Equation (15):

$$\Pr\left[\frac{1}{s'_{hi}}r_{hi}>(1+\delta)\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}\right]<\exp\left(-\frac{\delta^2}{3}\cdot\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}\cdot s'_{hi}\right)<\frac{1}{4n^3},$$

and

$$\Pr\left[\frac{1}{s'_{hi}}r_{hi}<(1-\delta)\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}\right]<\exp\left(-\frac{\delta^2}{2}\cdot\frac{\Delta_{hi}(v)}{|\Gamma_{hi}(v)|^2}\cdot s'_{hi}\right)<\frac{1}{4n^3}.$$

Hence, if $\Delta_{hi}(v)\geq\delta\cdot\mu_{j-1}$, then with probability at least $1-\frac{1}{n^3}$,

$$(1-\delta)\Delta_{hi}(v)\leq\widehat{\Delta}_{hi}\leq(1+\delta)\Delta_{hi}(v), \tag{16}$$

and the proof is complete for the case that $\Delta_{hi}(v)\geq\delta\cdot\mu_{j-1}$.

Now consider a vertex v such that $\Delta_{hi}(v)<\delta\cdot\mu_{j-1}$. If $|\Gamma_{hi}(v)|\geq\sqrt{\delta\mu_{j-1}}$, then a similar analysis to the one above gives that with probability at least $1-\frac{1}{n^3}$,

$$\Pr\left[\frac{1}{s'_{hi}}\cdot r_{hi}>(1+\delta)\frac{\delta\mu_{j-1}}{|\Gamma_{hi}(v)|^2}\right]<\exp\left(-\frac{\delta^2}{3}\cdot\frac{\delta\mu_{j-1}}{|\Gamma_{hi}(v)|^2}\cdot s'_{hi}\right)<\frac{1}{4n^3}.$$

Therefore if $\Delta_{hi}(v)<\delta\mu_{j-1}$, then the procedure returns $\widehat{\Delta}_{hi}$ such that $\widehat{\Delta}_{hi}<(1+\delta)\cdot\delta\mu_{j-1}$, with probability at least $1-\frac{1}{n^3}$, as required. ■

Claim 3.3.7 *For every vertex v and every index $j\in[k']$, the expected query complexity and running time of the procedure* Approx-High-Triangles$(v,j,d(v),\delta)$ *is* $\frac{d(v)\cdot\sqrt{m}}{\mu_j}\cdot\mathrm{poly}(\log n,1/\epsilon)$.

Proof: The procedure performs $s = \frac{d(v)}{\sqrt{\delta \cdot \mu_{j-1}}} \cdot \frac{20 \log n}{\delta^2}$ queries in Step (1). If $\widehat{\Gamma}_{hi} > (1-\delta)\sqrt{\delta \cdot \mu_{j-1}}$, then at most $3s' = 3 \cdot \frac{\widehat{\Gamma}_{hi} \cdot d(v)}{(1-\delta)^2 \cdot \delta \cdot \mu_{j-1}} \cdot \frac{30 \log n}{\delta^2}$ additional queries are performed. Therefore the query complexity of the algorithm is $\Theta(s + s')$. By the setting of $k' = \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1)}{\beta} \overline{\mathcal{T}}^{2/3}$, Equation (3) and Item (3) in Assumption 2.1, for every $j \in [k']$,

$$\mu_j \leq (1+\beta)^{k'} = \frac{c_\Delta \cdot (k+1)}{\beta} \overline{\mathcal{T}}^{2/3} \leq \frac{c_\Delta \cdot (k+1)}{\beta} \cdot (3\overline{\Delta})^{2/3} \leq \frac{3c_\Delta \cdot (k+1)}{\beta} \cdot 3\overline{m} \ .$$

It follows that:

$$\begin{aligned} s &= \frac{d(v)}{\sqrt{\delta \cdot \mu_{j-1}}} \cdot \frac{20 \log n}{\delta^2} \leq \frac{1}{\sqrt{\delta}} \cdot \frac{d(v)\sqrt{\overline{m}}}{\mu_{j-1}} \cdot \sqrt{\frac{9c_\Delta \cdot (k+1)}{\beta}} \cdot \frac{20 \log n}{\delta^2} \\ &= \frac{d(v) \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon) \ . \end{aligned} \tag{17}$$

It follows from the proof of Claim 3.3.6 that $\mathbb{E}[\widehat{\Gamma}_{hi}] = |\Gamma_{hi}(v)|$, and that with probability at least $1 - \frac{1}{4n^3}$, $\widehat{\Gamma}_{hi} < (1+\delta)\Gamma_{hi}(v)$. Also, by the Item (2) in Assumption 2.1,

$$|\Gamma_{hi}(v)| \leq |V_{hi}| \leq \frac{2m}{\sqrt{\overline{m}}} \leq \frac{2c_m \cdot \overline{m}}{\sqrt{\overline{m}}} \leq 2c_m \cdot \sqrt{\overline{m}}. \tag{18}$$

Therefore with probability at least $1 - \frac{1}{4n^3}$,

$$s' \leq \frac{\widehat{\Gamma}_{hi} \cdot d(v)}{(1-\delta)^2 \cdot \delta \cdot \mu_{j-1}} \cdot \frac{400 \log n}{\delta^2} = \frac{d(v) \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon). \tag{19}$$

The claim follows from Equations (17) and (19). ■

Algorithm 4 Approx-Crossing-Triangles$(v, j, d(v), \delta)$

1: $r_{cr} \leftarrow 0$.
2: **Repeat** $s_{cr} = \frac{d(v)\sqrt{\overline{m}}}{\delta \cdot \mu_{j-1}} \cdot \frac{20 \log n}{\delta^2}$ times:
3: Uniformly independently at random choose $u \in \Gamma(v)$ and query $d(u)$.
4: Uniformly independently at random choose $\ell \in [1, \ldots, \sqrt{\overline{m}}]$.
5: **If** $u \in \Gamma_{\ell o}(v)$ and $\ell \leq d(u)$ **then**
6: Query for the ℓ^{th} neighbor of u. Denote it by u'.
7: Query $d(u')$.
8: **If** $u' \in \Gamma_{\ell o}(v)$ **and** $(u', v) \in E$ **then**
9: $r_{cr} \leftarrow r_{cr} + 1$.
10: **Else if** $u' \in \Gamma_{hi}(v)$ **and** $(u', v) \in E$
11: $r_{cr} \leftarrow r_{cr} + \frac{1}{2}$.
12: Let $\widehat{\Delta}_{cr} \leftarrow \frac{r_{cr}}{s_{cr}} \cdot d(v)\sqrt{\overline{m}}$
13: **Return** $\widehat{\Delta}_{cr}$.

Definition 3.3.5 *We say that* Approx-Crossing-Triangles$(v, j, d(v), \delta)$ *answers correctly if the procedure returns* $\widehat{\Delta}_{cr}$ *for which the following holds. If* $\Delta_{cr}(v) \geq \delta \cdot \mu_{j-1}$*, then* $\widehat{\Delta}_{cr}$ *such that* $(1-\delta)\Delta_{cr}(v) \leq \widehat{\Delta}_{cr} \leq (1+\delta)\Delta_{cr}(v)$ *and otherwise* $\widehat{\Delta}_{cr} \leq (1+\delta)\cdot\delta\cdot\mu_{j-1}$.

Claim 3.3.8 *For every* v *and for every* $j \in [k']$*, with probability at least* $1 - \frac{1}{n^3}$*, the procedure* Approx-Crossing-Triangles$(v, j, d(v), \delta)$ *answers correctly, as defined in Definition 3.3.5.*

Proof: Denote by $\mathrm{Tr}^1_{cr}(v)$ the set of triangles (v, u, u') such that both u and u' are in $V_{\ell o}$, and by $\mathrm{Tr}^2_{cr}(v)$ the set of triangles (v, u, u') such that either u or u', but not both, are in $V_{\ell o}$. Let $\Delta^1_{cr}(v) = |\mathrm{Tr}^1_{cr}(v)|$, $\Delta^2_{cr}(v) = |\mathrm{Tr}^2_{cr}(v)|$ and let $\chi_1, \ldots, \chi_{s_c}$ be Bernoulli random variables such that

$$\chi_i = \begin{cases} 1 & \text{if a triangle } (v,u,u') \in \mathrm{Tr}^1_{cr}(v) \text{ is sampled in the } i^{\text{th}} \text{ iteration} \\ \frac{1}{2} & \text{if a triangle } (v,u,u') \in \mathrm{Tr}^2_{cr}(v) \text{ is sampled in the } i^{\text{th}} \text{ iteration} \\ 0 & \text{otherwise} \end{cases}.$$

Therefore,

$$\mathbb{E}[\chi_i] = \frac{1}{2}\Pr[\text{a triangle from } \mathrm{Tr}^2_{cr}(v) \text{ is sampled}] + \Pr[\text{a triangle from } \mathrm{Tr}^1_{cr}(v) \text{ is sampled}]. \quad (20)$$

We analyze the two terms separately.

$$\begin{aligned} &\Pr\left[\text{a specific triangle } (v,u,u') \in \mathrm{Tr}^2_{cr}(v) \text{ is sampled}\right] \\ &= \Pr[u \text{ is sampled in Step (3)}, u \in \Gamma_{\ell o}(v), \ell \leq d(u), u' \text{ is sampled in Step (6)}] \\ &+ \Pr[u' \text{ is sampled in Step (3)}, u' \in \Gamma_{\ell o}(v), \ell \leq d(u'), u \text{ is sampled in Step (6)}] \\ &= \frac{1}{d(v)} \cdot 1 \cdot \frac{d(u)}{\sqrt{m}} \cdot \frac{1}{d(u)} + \frac{1}{d(v)} \cdot 1 \cdot \frac{d(u')}{\sqrt{m}} \cdot \frac{1}{d(u')} = \frac{2}{d(v)\cdot\sqrt{m}}. \end{aligned}$$

Therefore,

$$\Pr\left[\text{some triangle in } \mathrm{Tr}^2_{cr}(v) \text{ is sampled}\right] = \frac{2\cdot\Delta^2_{cr}(v)}{d(v)\cdot\sqrt{m}}. \quad (21)$$

Now consider a triangle $(v, u, u') \in \mathrm{Tr}^1_{cr}(v)$. Without loss of generality, u is in $\Gamma_{\ell o}(v)$ (and u' is not). Hence,

$$\begin{aligned} &\Pr\left[\text{a specific triangle } (v,u,u') \in \mathrm{Tr}^1_{cr}(v) \text{ is sampled}\right] \\ &= \Pr[u \text{ is sampled in Step (3)}, u \in \Gamma_{\ell o}(v), \ell \leq d(u), u' \text{ is sampled in Step (6)}] \\ &= \frac{1}{d(v)} \cdot 1 \cdot \frac{d(u)}{\sqrt{m}} \cdot \frac{1}{d(u)} = \frac{1}{d(v)\cdot\sqrt{m}}, \end{aligned}$$

and

$$\Pr\left[\text{any triangle} \in \mathrm{Tr}^1_{cr}(v) \text{ is sampled}\right] = \frac{\Delta^1_{cr}(v)}{d(v)\cdot\sqrt{m}}. \quad (22)$$

Plugging Equation (21) and Equation (22) into Equation (20) we get:

$$\mathbb{E}[\chi_i] = \frac{1}{2}\cdot\frac{2\cdot\Delta^2_{cr}(v)}{d(v)\cdot\sqrt{m}} + \frac{\Delta^1_{cr}(v)}{d(v)\cdot\sqrt{m}} = \frac{\Delta_{cr}(v)}{d(v)\cdot\sqrt{m}}.$$

Note that by the definition of the χ_i variables, $r_{cr} = \sum_{i=1}^{s_{cr}} \chi_i$. If $\Delta_{cr}(v) \geq \delta \cdot \mu_{j-1}$ then by applying the multiplicative Chernoff bound and by our choice of s_{cr}, we get that:

$$\begin{aligned}
\Pr\left[\frac{1}{s_{cr}} r_{cr} > (1+\delta)\frac{\Delta_{cr}(v)}{d(v)\cdot\sqrt{m}}\right] &< \exp\left(-\frac{\delta^2}{3}\cdot\frac{\Delta_{cr}(v)}{d(v)\cdot\sqrt{m}}\cdot s_{cr}\right) \\
&< \exp\left(-\frac{\delta^2}{3}\cdot\frac{\Delta_{cr}(v)}{d(v)\cdot\sqrt{m}}\cdot\frac{d(v)\cdot\sqrt{m}}{\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2}\right) \\
&< \frac{1}{2n^3}.
\end{aligned}$$

Similarly $\Pr\left[\frac{1}{s_{cr}} r_{cr} < (1-\delta)\frac{\Delta_{cr}(v)}{d(v)\cdot\sqrt{m}}\right] < \frac{1}{2n^3}$. Since $\widehat{\Delta}_{cr} = \frac{r_{cr}}{s_{cr}} \cdot d(v)\sqrt{m}$ with probability at least $1 - \frac{1}{n^3}$,

$$(1-\delta)\cdot\Delta_{cr}(v) \leq \widehat{\Delta}_{cr} \leq (1-\delta)\cdot\Delta_{cr}(v).$$

If $\Delta_{cr} < \delta \cdot \mu_{j-1}$ then

$$\begin{aligned}
\Pr\left[\frac{1}{s_{cr}} r_{cr} > (1+\delta)\frac{\delta\cdot\mu_{j-1}}{d(v)\cdot\sqrt{m}}\right] &< \exp\left(-\frac{\delta^2}{3}\cdot\frac{\delta\cdot\mu_{j-1}}{d(v)\cdot\sqrt{m}}\cdot s_{cr}\right) \\
&< \exp\left(-\frac{\delta^2}{3}\cdot\frac{\delta\cdot\mu_{j-1}}{d(v)\cdot\sqrt{m}}\cdot\frac{d(v)\cdot\sqrt{m}}{\delta\cdot\mu_{j-1}}\cdot\frac{20\log n}{\delta^2}\right) \\
&< \frac{1}{2n^3},
\end{aligned}$$

and the proof is complete. ■

Claim 3.3.9 *For every vertex v and every index $j \in [k']$, the query complexity and running time of the procedure* Approx-Crossing-Triangles$(v, j, d(v), \delta)$ *are* $\frac{d(v)\cdot\sqrt{m}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon)$.

Proof: In each iteration of the loop in Step (2) the procedure performs at most 5 queries: two neighbor queries u, u', two degree queries $d(u), d(u')$ and a pair query (u, u'). In all other steps the running time is constant and no queries are performed. Therefore the query complexity and running time are bounded by

$$5 \cdot s_{cr} = 5 \cdot \frac{d(v)\sqrt{m}}{\delta\cdot\mu_{j-1}} \cdot \frac{20\log n}{\delta^2} = \frac{d(v)\cdot\sqrt{m}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon),$$

as required. ■

We are now ready to present the procedure for estimating the triangles-degree of a vertex.

Procedure 5 Approx-Triangles-Degree (v, j)

Let $\delta \leftarrow \frac{1}{7} \cdot \frac{\gamma}{\log n} \cdot \frac{\beta}{1+\beta}$.
Query v's degree $d(v)$.
If $d(v) \leq \sqrt{m}$ **then**
    $\widehat{\Delta} \leftarrow$ Approx-Triangles-Of-Low-Deg-Vertices$(v, j, d(v), \delta)$.
Else
    $\widehat{\Delta}_{hi} \leftarrow$ Approx-High-Triangles$(v, j, d(v), \delta)$.
    $\widehat{\Delta}_{cr} \leftarrow$ Approx-Crossing-Triangles$(v, j, d(v), \delta)$.
    $\widehat{\Delta} \leftarrow \widehat{\Delta}_{hi} + \widehat{\Delta}_{cr}$.
Return $\widehat{\Delta}$.

Definition 3.3.6 *Let δ be as defined in Approx-Triangles-Degree. We say that Approx-Triangles-Degree(v, j) answers correctly in the following cases.*

- *If $\Delta(v) \geq \delta \cdot \mu_{j-1}$ and Approx-Triangles-Degree(v, j) returns $\widehat{\Delta}$ such that $(1 - 3\delta)\Delta(v) \leq \widehat{\Delta} \leq (1 + 3\delta)\Delta(v)$.*
- *If $\Delta(v) < \delta \cdot \mu_{j-1}$ and the procedure returns $\widehat{\Delta}$ such that $\widehat{\Delta} \leq (1 + \delta) \cdot 2\delta \cdot \mu_{j-1}$.*

Lemma 3.3.10

1. *For every v and every $j \in [k']$, Approx-Triangles-Degree(v, j) answers correctly with probability at least $1 - \frac{2}{n^3}$.*
2. *For every v and every $j \in [k']$, the expected query complexity and running time of the procedure are $\frac{d(v) \cdot \sqrt{m}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon)$.*

Proof: The second item of the lemma is a direct corollary of Claims 3.3.5, 3.3.7 and 3.3.9. Therefore, it remains to prove the first item.

Consider a vertex $v \in V_{\ell o}$. If $\Delta(v) \geq \delta\mu_{j-1}$ then by Claim 3.3.4, it holds that $(1 - \delta)\Delta(v) \leq \widehat{\Delta} \leq (1 + \delta)\Delta(v)$ with probability at least $1 - \frac{1}{n^3}$. Otherwise, if $\Delta(v) < \delta\mu_{j-1}$, then Approx-Deg-Of-Low-Deg-Vertices returns $\widehat{\Delta}$ such that $\widehat{\Delta} \leq (1 + \delta) \cdot \delta \cdot \mu_{j-1}$ with probability at least $1 - \frac{1}{n^3}$. In either case the lemma follows.

Now consider a vertex $v \in V_{hi}$. If $\Delta(v) \geq \mu_{j-1}$, then either $\Delta_{hi}(v)$ or $\Delta_{cr}(v)$ are at least $\frac{1}{2}\mu_{j-1}$. Assume first that both $\Delta_{hi}(v)$ and $\Delta_{cr}(v)$ are at least $\delta \cdot \mu_{j-1}$. By the first part of Claim 3.3.6 and the first part of Claim 3.3.8 it holds that with probability at least $1 - \frac{1}{n^3}$,

$$(1 - \delta)\Delta_{hi}(v) \leq \widehat{\Delta}_{hi} \leq (1 + \delta)\Delta_{hi}(v) \tag{23}$$

and with probability at least $1 - \frac{1}{n^3}$,

$$(1 - \delta)\Delta_{cr}(v) \leq \widehat{\Delta}_{cr} \leq (1 + \delta)\Delta_{cr}(v). \tag{24}$$

By Equations (23) and (24) and by the union bound we get that with probability at least $1 - \frac{2}{n^3}$,

$$(1 - \delta)\Delta(v) \leq \widehat{\Delta} \leq (1 + \delta)\Delta(v),$$

as required.

We now turn to the case that either $\Delta_{hi}(v)$ or $\Delta_{cr}(v)$ are smaller than $\delta\mu_{j-1}$, but not both. Assume without loss of generality that $\Delta_{hi}(v) < \delta\mu_{j-1}$. By Claim 3.3.6, we have that $\widehat{\Delta}_{hi} \leq (1+\delta)\cdot\delta\cdot\mu_{j-1}$ with probability at least $1-\frac{1}{n^3}$, and by Claim 3.3.8, we have that $(1-\delta)\Delta_{cr}(v) \leq \widehat{\Delta}_{cr} \leq (1+\delta)\Delta_{cr}(v)$ with probability at least $1-\frac{1}{n^3}$. Therefore, with probability at least $1-\frac{2}{n^3}$,

$$\widehat{\Delta} = \widehat{\Delta}_{hi} + \widehat{\Delta}_{cr} \leq (1+\delta)\Delta_{cr}(v) + (1+\delta)\cdot\delta\cdot\mu_{j-1} < (1+3\delta)\Delta(v). \tag{25}$$

To prove the lower bound on $\widehat{\Delta}$ note that since $\Delta_{cr}(v) = \Delta(v) - \Delta_{hi}(v)$ and $\Delta_{hi}(v) < \delta\mu_{j-1}$, it holds that $\Delta_{cr}(v) \geq \Delta(v) - \delta\mu_{j-1}$. Therefore,

$$\widehat{\Delta} \geq \widehat{\Delta}_{cr} \geq (1-\delta)\Delta_{cr}(v) \geq (1-\delta)(\Delta(v) - \delta\mu_{j-1}) \geq (1-2\delta)\Delta(v). \tag{26}$$

Combining Equations (25) and (26) we get that if $\Delta(v) \geq \mu_{j-1}$ then with probability at least $1-\frac{2}{n^3}$.

$$(1-3\delta) \leq \Delta(v) \leq (1+3\delta)\Delta,$$

as required. If both $\Delta_{hi}(v)$ and $\Delta_{cr}(v)$ are smaller than $\delta\cdot\mu_{j-1}$, then $\Delta(v) < 2\delta\cdot\mu_{j-1}$. By Claim 3.3.6 and Claim 3.3.8, it holds that with probability at least $1-\frac{1}{n^3}$,

$$\widehat{\Delta}_{hi} \leq (1+\delta)\cdot\delta\cdot\mu_{j-1} \tag{27}$$

and with probability at least $1-\frac{1}{n^3}$,

$$\widehat{\Delta}_{cr} \leq (1+\delta)\cdot\delta\cdot\mu_{j-1}. \tag{28}$$

By Equations (27) and (28) and by the union bound we get that with probability at least $1-\frac{2}{n^3}$,

$$\widehat{\Delta} \leq (1+\delta)\cdot 2\delta\cdot\mu_{j-1},$$

and the proof of the first item is complete. ■

The following is a corollary of Item (1) in Lemma 3.3.10.

Corollary 3.3.11 *With probability at least* $1-\frac{2}{n^3}$*:*

1. *If* Approx-Triangles-Degree *is invoked with a vertex* v *and an index* j *such that* $v \in B'_j$ *then* Approx-Triangles-Degree *returns* $\widehat{\Delta}(v)$ *such that* $\widehat{\Delta}(v) \in (\mu_{j-1}, \mu_j]$.
2. *If* Approx-Triangles-Degree *is invoked with a vertex* v *and an index* j *such that* $v \notin B''_j$ *then* Approx-Triangles-Degree *returns* $\widehat{\Delta}(v)$ *such that* $\widehat{\Delta}(v) \notin (\mu_{j-1}, \mu_j]$.

3.3.3 Significant buckets

Recall that by Lemma 3.3.10 the query complexity and running time of the procedure Approx-Triangles-Degree(v,j) are roughly $O\left(\frac{d(v)\cdot\sqrt{m}}{\mu_j}\right)$, and consider using the procedure instead of the oracle for $\Delta(v)$. That is, we select a sample of $\Theta\left(\frac{n}{T^{1/3}}\right)\cdot\mathrm{poly}(\log n, 1/\epsilon)$ vertices, and for each

selected vertex v, we run the procedure for decreasing values of j, until we find an index j such that $\widehat{\Delta} \in (\mu_{j-1}, \mu_j]$. Putting aside the fact that the procedure only provides an estimate of $\Delta(v)$ (so that v may be assigned to the wrong bucket), the complexity of the resulting algorithm may be much higher than the complexity we are aiming for: $\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$. This is due to vertices v whose triangles-degree $\Delta(v)$ is relatively small (so that we need to run the procedure with small j), while their (neighbors) degree $d(v)$ is relatively large.

In order to reduce the overall complexity of the algorithm we first make the following observation. Recall that by the definition of the strict buckets, the number of labeled triangles that have a vertex in B'_j is roughly $|B'_j| \cdot \mu_j$. Therefore, if $|B'_j| \cdot \mu_j$ is relatively small, i.e., smaller than $\frac{\beta|\mathcal{T}(G)|}{k+1}$, then we can disregard this bucket (that is, assume it is empty). This means that we only need to estimate the sizes of strict buckets B'_j such that $|B'_j| \geq \frac{\beta|\mathcal{T}(G)|}{(k+1)\cdot\mu_j}$. In order to "hit" such a bucket and estimate its size, it suffices to take a sample whose size grows (roughly) like $\frac{n}{|B'_j|}$ which is at most $\frac{n\cdot(k+1)\cdot\mu_j}{\beta|\mathcal{T}(G)|}$.

We thus see that while the complexity of Approx-Triangles-Degree increases as j decreases, the size of the sample sufficient for estimating the size of B'_j decreases as j decreases. Indeed, the product of the two does not depend on j. However, this product does depend on $d(v)$, which may be large. Our second observation is that for any degree d, the number of vertices with degree greater than d is upper bounded by $2m/d$. If we take a sample of s vertices, then we do not expect to get many more than $\frac{s}{n} \cdot \frac{2m}{d}$ vertices with degree greater than d. So while the complexity of Approx-Triangles-Degree increases with the degree of the vertex it is given, the number of vertices with at least any given degree, decreases with the degree.

In order to benefit from the above tradeoffs, we introduce the notion of significance. We note that in what follows, unless explicitly stated otherwise, when we say "buckets" we refer to strict buckets as defined in Equation (9).

Definition 3.3.7 *For a set of indices $X \subseteq [k]$, let B'_X denote the union of all the strict buckets B'_j for which $j \in X$. That is, $B'_X = \bigcup_{j\in X} B'_j$.*

Definition 3.3.8 *For sets on indices $X, Y, Z \subseteq [k]$, let $Tr_{X,Y,Z}$ denote the set of triangles (u, v, w) such that $u \in B'_X$, $v \in B'_Y$ and $w \in B'_W$.*

Definition 3.3.9 *We say that a bucket B'_j is significant if $|B'_j| \geq \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}$, and otherwise we say it is insignificant.*

Note that for every $j \in [k]$, $|\mathcal{T}(B'_j)|$ is at most $|B'_j| \cdot \mu_j$. Therefore if a bucket B'_j is insignificant, then $|\mathcal{T}(B'_j)| < \frac{\beta\overline{\mathcal{T}}}{k+1}$, implying that the bucket's contribution to the total number of triangles in the graph is insignificant.

We now redefine our notion of largeness to include both largeness in size and in contribution to the number of triangles.

Definition 3.3.10 *Let $\mathcal{L}^* = \left\{j \in [k] : |B'_j| \geq \max\left\{\frac{(\beta\cdot\overline{\mathcal{T}})^{1/3}}{k+1}, \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}\right\}\right\}$. That is, $\mathcal{L}^*$ is the set of indices of large significant buckets.*

To see why approximating the sizes of the large significant buckets gives a good approximation of $|\mathcal{T}(G)|$, we prove that similar claims to Claim 3.2.1 and Claim 3.2.2 hold for the redefined set $\mathcal{L}^*$.

Definition 3.3.11 *Let Tr_I denote the set of triangles with an endpoint in B_{I_j} for some $j \in [k]$ or with an endpoint in an insignificant bucket.*

Claim 3.3.12 *If the selection of the safety intervals is good (as defined in Definition 3.3.1), then for $\mathcal{L}^*$ as defined in Definition 3.3.10,*

$$\frac{1}{3}(1-30\beta)|\mathcal{T}(G)| \leq |\mathcal{T}(B'_{\mathcal{L}^*})| \leq |\mathcal{T}(G)|.$$

Proof: As in the proof of Claim 3.2.2, we consider the following partition of the graph's triangles into disjoint sets: $Tr_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{L}^*}$, $Tr_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}$, $Tr_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}$, $Tr_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}$ and Tr_I. Since the above sets are disjoint, we have that:

$$|\mathcal{T}(G)| = 3|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}| + 3|\mathrm{Tr}_{\mathcal{L},\mathcal{S}^*,\mathcal{S}^*}| + 3|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| + 3|\mathrm{Tr}'_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}| + 3|\mathrm{Tr}_I|, \tag{29}$$

and

$$\begin{aligned}|\mathcal{T}(B'_{\mathcal{L}^*})| &= 3|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{L}^*}| + 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| + |\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| \\ &= |\mathcal{T}(G)| - |\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| - 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| - 3|\mathrm{Tr}_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}| - 3|\mathrm{Tr}_I|\,. \end{aligned} \tag{30}$$

By Equation (29),

$$2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| + |\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| \leq 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{S}^*,\mathcal{S}^*}| + 2|\mathrm{Tr}_{\mathcal{L}^*,\mathcal{L}^*,\mathcal{S}^*}| \leq \frac{2}{3}|\mathcal{T}(G)|. \tag{31}$$

Observe that the set $\mathrm{Tr}_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}$ is a subset of the set of triangles with all three endpoints in small buckets. Hence, from Claim 3.2.1, it holds that

$$|\mathrm{Tr}_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}| \leq \beta|\mathcal{T}(G)|. \tag{32}$$

From the definition of a good selection of the safety intervals in Definition 3.3.1,

$$\sum_{j\in[k]} |B_{I_j}| \cdot \mu_j \leq \sum_{j\in[k]} \beta|\widetilde{B}_j|(1+\beta)^j \leq \beta|\mathcal{T}(G)|. \tag{33}$$

By the definition of the insignificant buckets and $\mathcal{T}(B_{\mathcal{I}})$, and by Equation (4),

$$|\mathcal{T}(B_{\mathcal{I}})| \leq \sum_{j\in\mathcal{I}} |B_j|\mu_j < \sum_{j\in\mathcal{I}} \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j} \cdot \mu_j \leq \beta\overline{\mathcal{T}} \leq \beta|\mathcal{T}(G)|. \tag{34}$$

It follows from Equation (33) and Equation (34) that

$$|\mathrm{Tr}_I| \leq \sum_{j\in[k]} |B_{I_j}| \cdot \mu_j + \mathcal{T}(B_I) \leq 2\beta|\mathcal{T}(G)|. \tag{35}$$

Plugging Equations (31), (32) and (35) into Equation (30) we get:

$$|\mathcal{T}(B'_{\mathcal{L}^*})| \geq |\mathcal{T}(G)| - \frac{2}{3}|\mathcal{T}(G)| - 3\beta|\mathcal{T}(G)| - 6\beta|\mathcal{T}(G)| = \frac{1}{3}(1-30\beta)|\mathcal{T}(G)|,$$

and the claim follows. ■

To complete the discussion concerning significant large buckets observe that any bucket with a high triangles-degree of $\Omega(\overline{\mathcal{T}}^{2/3})$ cannot be large, and that buckets with a low triangles-degree which are small are not significant. Therefore we can consider only significant buckets with a low triangles-degree. Formally, consider the following definition and claim.

Definition 3.3.12 *Let* $k' = \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1) \cdot \overline{\mathcal{T}}^{2/3}}{\beta}$, *and let*

$$\mathcal{L} = \left\{ j \in [k'] : \ |B'_j| \geq \frac{\beta \cdot \overline{\mathcal{T}}}{(k+1) \cdot \mu_j} \right\}.$$

Claim 3.3.13 *Let* $\mathcal{L}^*$ *be as defined in Definition 3.3.9, and let* k' *and* $\mathcal{L}$ *be as defined in Definition 3.3.12. It holds that* $\mathcal{L}^* \subseteq \mathcal{L}$.

Proof: Clearly for every j such that $|B'_j| \geq \max\left\{ \frac{(\beta \cdot \overline{\mathcal{T}})^{1/3}}{k+1}, \frac{\beta \cdot \overline{\mathcal{T}}}{(k+1) \cdot \mu_j} \right\}$ it holds that $|B'_j| \geq \frac{\beta \cdot \overline{\mathcal{T}}}{(k+1) \cdot \mu_j}$. Therefore, it remains to prove that for every $j \in \mathcal{L}^*$ it holds that $j \leq k'$. Assume towards a contradiction that there exists an index $j \in \mathcal{L}^*$ such that $j > k'$. It follows that $\mu_j > \frac{c_\Delta \cdot (k+1) \cdot \overline{\mathcal{T}}^{2/3}}{\beta}$, and that $|B'_j| \geq \frac{(\beta \cdot \overline{\mathcal{T}})^{1/3}}{k+1}$. The above, together with Equation (4), implies that:

$$|B'_j| \cdot \mu_j > \frac{(\beta \cdot \overline{\mathcal{T}})^{1/3}}{k+1} \cdot \frac{c_\Delta \cdot (k+1) \cdot \overline{\mathcal{T}}^{2/3}}{\beta} = \frac{c_\Delta \cdot \overline{\mathcal{T}}}{\beta^{2/3}} > c_\Delta \cdot \overline{\mathcal{T}} \geq |\mathcal{T}(G)|,$$

which is a contradiction, since a bucket cannot contribute more labeled triangles then there are in the graph. This completes the proof. ■

The following is a corollary of Claim 3.3.12 and Claim 3.3.13.

Corollary 3.3.14

$$\frac{1}{3}(1 - 30\beta)|\mathcal{T}(G)| \leq |\mathcal{T}(B'_{\mathcal{L}})| \leq |\mathcal{T}(G)|.$$

It follows from the above discussion that to get a $\frac{1}{3}$-approximation of $|\mathcal{T}(G)|$, it is sufficient to estimate the sizes of the significant buckets B'_j for j's such that $j \leq \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1) \cdot \overline{\mathcal{T}}^{2/3}}{\beta}$.

3.3.4 The algorithm for a $\frac{1}{3}$-approximation

We are now ready to present our algorithm for $\frac{1}{3}$-approximating $|\mathcal{T}(G)|$.

Procedure 6 $\frac{1}{3}$-Approx-Triangles $(G, \overline{\mathcal{T}}, \overline{m}, \epsilon)$

1: Let $\beta = \epsilon/450$, $k' = \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1)}{\beta} \overline{\mathcal{T}}^{2/3}$ and $\gamma = \frac{\beta}{4 \log^3 n}$. ▷ Where c_Δ is a constant that will be set later on.
2: Let $\{\mu_0, \ldots, \mu_{k'}\} \leftarrow$ Create-Random-Thresholds(β, γ, k').
3: **For** $j = 0, \ldots, k'$ **do**
4: $\quad \widehat{t}_j \leftarrow 0$.
5: $\quad$ Uniformly and independently select $s_j = \frac{n \cdot \mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n \cdot (k+1)}{\beta^3}$ vertices.
6: $\quad$ Denote by $S_j = \{v_{j,1}, \ldots, v_{j,s_j}\}$ the multiset of selected vertices.
7: $\quad$ **For each** v in S_j **do**
8: $\quad\quad \widehat{\Delta}(v) \leftarrow$ Approx-Triangles-Degree(v, j).
9: $\quad\quad$ **If** $\widehat{\Delta}(v) \in (\mu_{j-1}, \mu_j]$ **then**
10: $\quad\quad\quad \widehat{t}_j \leftarrow \widehat{t}_j + 1$.
11: $\quad \widehat{b}_j \leftarrow \widehat{t}_j \cdot \frac{n}{s_j}$.
12: Let $\widehat{\mathcal{L}} = \{j : \widehat{b}_j \geq (1-\beta)\frac{\beta \cdot \overline{\mathcal{T}}}{k' \cdot \mu_j}\}$.
13: **Return** $\widehat{t} = \sum_{j \in \widehat{\mathcal{L}}} \widehat{b}_j \cdot \mu_j$.

Theorem 3.2 *Algorithm* $\frac{1}{3}$*-Approx-Triangles returns* $\widehat{t}$ *such that, with probability at least* $1 - \frac{1}{2 \log^3 n}$,

$$\frac{1}{3}(1 - 50\beta)|\mathcal{T}(G)| \leq \widehat{t} \leq (1 + 50\beta)|\mathcal{T}(G)|.$$

In order to prove Theorem 3.2 we first establish the following claim and corollary.

We note that we delay the discussion regarding the running time of the algorithm for Subsection 3.5, where it will be analyzed as part of the $(1 \pm \epsilon)$-approximation algorithm.

Definition 3.3.13 *For every* $j \in [k']$, *let* $\widehat{b}_j$ *be as defined in Step* (11) *of* $\frac{1}{3}$*-Approx-Triangles. We say that the algorithm estimates the buckets' sizes correctly, if the following holds.*

1. *For every* $j \in \mathcal{L}$, *it holds that* $(1-\beta)|B'_j| \leq \widehat{b}_j \leq (1+\beta)|B''_j|$.
2. *For every* $j \notin \mathcal{L}$, *it holds that* $0 \leq \widehat{b}_j < (1+\beta) \cdot \left(\frac{\beta \overline{\mathcal{T}}}{(k+1)\mu_j} + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|\right)$.

Claim 3.3.15 *If the output of Create-Random-Thresholds is good (as Defined in Definition 3.3.1) and Approx-Triangles-Degree*(v, j) *answers correctly on all the sampled vertices (as defined in Definition 3.3.6), then* $\frac{1}{3}$*-Approx-Triangles estimates the buckets' sizes correctly, with probability at least* $1 - \frac{1}{n^2}$ *over the choices of the samples* S_j.

Proof: Let $\chi'_{j,i}$ for $j \in [k'], 1 \leq i \leq s_j$ be Bernoully random variables such that $\chi'_{j,i} = 1$ if and only if $v_{j,i} \in B'_j$, so that $\sum_{i=1}^{s_j} \chi'_{j,i} = |S_j \cap B'_j|$. By the definition of $\chi_{j,i}$

$$\Pr[\chi'_{j,i} = 1] = \frac{|B'_j|}{n}.$$

Consider an index $j \in \mathcal{L}$. By the definition of $\mathcal{L}$, it holds that $|B'_j| \geq \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}$. By the selection of s_j in the algorithm,

$$s_j \cdot \frac{|B'_j|}{n} \geq \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j} \cdot \frac{1}{n} \cdot \frac{n\cdot\mu_j}{\overline{\mathcal{T}}} \cdot \frac{20\log n\cdot(k+1)}{\beta^3} = \frac{20\log n}{\beta^2}\ .$$

Let $\widehat{t'}_j = \sum_{i=1}^{s_j} \chi'_{j,i}$. By applying the multiplicative Chernoff bound we get:

$$\Pr\left[\frac{1}{s_j}\widehat{t'}_j < (1-\beta)\frac{|B'_j|}{n}\right] < \Pr\left[\frac{1}{s_j}\widehat{t'}_j < (1-\beta)\frac{|B'_j|}{n}\right] < \exp\left(-\frac{\beta^2}{2}\cdot\frac{|B'_j|}{n}\cdot s_j\right) < \frac{1}{2n^3}. \tag{36}$$

By the assumption that Approx-Triangles-Degree(v,j) answers correctly on all the sampled vertices, and by Corollary 3.3.11, we have that if $v \in B'_j$, then $\widehat{\Delta}$ is such that $\widehat{\Delta} \in (\mu_{j-1},\mu_j]$. Therefore, for every sampled vertex from B'_j, $\widehat{t}_j$ is incremented, and it follows that $\widehat{t}_j \geq \widehat{t'}_j$.

To upper bound $\widehat{t}_j$ consider the following. Let $\chi''_{j,i}$ for $j \in [k']$, $1 \leq i \leq s_j$ be a Bernoully random variable such that $\chi''_{j,i} = 1$ if and only if $v_{j,i} \in B''_j$, so that $\sum_{i=1}^{s_j} \chi''_{j,i} = |S_j \cap B''_j|$. By the definition of $\chi''_{j,i}$,

$$\Pr[\chi''_{j,i} = 1] = \frac{|B''_j|}{n}.$$

Let $\widehat{t''}_j = \sum_{i=1}^{s_j} \chi''_{j,i}$. By applying the multiplicative Chernoff bound we get:

$$\Pr\left[\frac{1}{s_j}\widehat{t''}_j > (1+\beta)\frac{|B''_j|}{n}\right] < \exp\left(-\frac{\beta^2}{3}\cdot|B''_j|\cdot\frac{s_j}{n}\right) < \exp\left(-\frac{\beta^2}{3}\cdot|B'_j|\cdot\frac{s_j}{n}\right) < \frac{1}{2n^3}. \tag{37}$$

By Corollary 3.3.11, for any vertex v not in B''_j, Approx-Triangles-Degree(v,j) returns $\widehat{\Delta}$ such that $\widehat{\Delta} \notin (\mu_{j-1},\mu_j]$. Hence we have that $\widehat{t}_j \leq \widehat{t''}_j$. By Equations (36) and (37), and by the previous discussion showing that $\widehat{t}_j \geq \widehat{t'}_j$, we get that for significant buckets B'_j, with probability at least $1-\frac{1}{n^3}$,

$$\frac{s_j}{n}\cdot(1-\beta)|B'_j| \leq \widehat{t}_j \leq \frac{s_j}{n}\cdot(1+\beta)(|B'_j| + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|). \tag{38}$$

Recalling that $\widehat{b}_j = \widehat{t}_j \cdot \frac{n}{s_j}$, this implies that for buckets B'_j such that $j \in \mathcal{L}$, with probability at least $1-\frac{1}{n^3}$,

$$(1-\beta)|B'_j| \leq \widehat{b}_j \leq (1+\beta)|B''_j|, \tag{39}$$

as stated in Item (1) of the current claim.

Now consider an index j such that $j \notin \mathcal{L}'$. From the assumption that the selection of the safety intervals by Create-Random-Thresholds was good, we have that

$$\Pr[\chi''_{j,i} = 1] = \frac{|B''_j|}{n} \leq \frac{|B'_j| + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|}{n} \leq \frac{\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j} + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|}{n}.$$

By applying the multiplicative Chernoff bound we get:

$$\Pr\left[\frac{1}{s_j}\widehat{t''}_j > (1+\beta)\frac{\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}+\beta|\widetilde{B}_j|+\beta|\widetilde{B}_{j-1}|}{n}\right] < \exp\left(-\frac{\beta^2}{3}\cdot\left(\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}+\beta|\widetilde{B}_j|+\beta|\widetilde{B}_{j-1}|\right)\cdot\frac{s_j}{n}\right)$$
$$< \exp\left(-\frac{\beta^2}{3}\cdot\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}\cdot\frac{s_j}{n}\right) < \frac{1}{2n^3}. \qquad (40)$$

Therefore, for buckets B'_j such that $j \in [k'] \setminus \mathcal{L}$, we have that, with probability at least $1-\frac{1}{2n^3}$,

$$\widehat{b}_j \leq (1+\beta)\left(\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}+\beta|\widetilde{B}_j|+\beta|\widetilde{B}_{j-1}|\right).$$

By taking the union bound over all the buckets, we get that the claim holds for every $j \in [k']$ with probability at least $1-\frac{1}{n^2}$. ■

The following is a corollary of Item (1) in Definition 3.3.13 and Claim 3.3.13.

Corollary 3.3.16 *Let $\mathcal{L}^*$ be as defined in Definition 3.3.10, $\mathcal{L}$ as defined in Definition 3.3.12 and $\widehat{\mathcal{L}}$ as defined in Step (12) of $\frac{1}{3}$-Approx-Triangles. If $\frac{1}{3}$-Approx-Triangles estimates the buckets' sizes correctly, then*

$$\widehat{\mathcal{L}} \supseteq \mathcal{L} \supseteq \mathcal{L}^*.$$

Proof: Consider an index $j \in \mathcal{L}$. By the definition of $\mathcal{L}$, B'_j is significant, and therefore, by the assumption that $\frac{1}{3}$-Approx-Triangles estimates the buckets' sizes correctly, it holds that

$$\widehat{b}_j \geq (1-\beta)|B'_j| \geq (1-\beta)\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}.$$

Hence, $j \in \widehat{\mathcal{L}}$. By Claim 3.3.13, $\mathcal{L} \supseteq \mathcal{L}^*$ and the corollary follows. ■

Corollary 3.3.17 *If $\frac{1}{3}$-Approx-Triangles estimates the buckets' sizes correctly, then for every $j \in \widehat{\mathcal{L}}$ (where $\widehat{\mathcal{L}}$ is as defined in Step (12) of the algorithm), $\widehat{b}_j \geq (1-\beta)|B'_j|$.*

Proof: For significant buckets the corollary follows directly from the definition of estimating the bucket's sizes correctly. Therefore, assume towards a contradiction that there exists an insignificant bucket B'_j such that $j \in \widehat{\mathcal{L}}$ and for which $\widehat{b}_j < (1-\beta)|B'_j|$. By the definition of $\widehat{\mathcal{L}}$,

$$\widehat{b}_j \geq (1-\beta)\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j},$$

implying that

$$|B'_j| > \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}.$$

Therefore This is a contradiction to assumption that B'_j is insignificant, and therefore the corollary follows. ■

Claim 3.3.18 *Algorithm* $\frac{1}{3}$*-Approx-Triangles returns* $\widehat{t}$ *such that, with probability at least* $1-\frac{1}{2\log^3 n}$,

$$(1-\beta)|\mathcal{T}(B'_{\mathcal{L}})| \leq \widehat{t} \leq (1+50\beta)|\mathcal{T}(B'_{\mathcal{L}})|.$$

Proof: We define the following set of "bad" events:

1. E_1 – The selection of the safety intervals in Create-Random-Thresholds was not a good selection, where a good selection is as defined in Definition 3.3.1.
2. E_2 – Approx-Triangles-Degree(v,j) did not answer correctly on some sampled vertex, where "answering correctly" is as defined in Definition 3.3.6.
3. E_3 – Algorithm $\frac{1}{3}$-Approx-Triangles did not estimate the buckets' sizes correctly, as defined in Definition 3.3.13.

By Claim 3.3.2 the probability of event E_1 occurring is at most $\frac{1}{4\log^3 n}$. By Lemma 3.3.10 and by taking the union bound over all the invocations of the Approx-Triangles-Degree procedure, if event E_1 does not occur, then event E_2 occurs with probability at most $\frac{2}{n^2}$. By Claim 3.3.15, if events E_1 and E_2 do not occur, then event E_3 occurs with probability at most $\frac{1}{n^2}$. Therefore any of these events occur with probability at most $\frac{1}{2\log^3 n}$, and the following discussion holds with probability at least $1-\frac{1}{2\log^3 n}$.

By the definition of $\widehat{t}$ in the algorithm,

$$\widehat{t} = \sum_{j\in\widehat{\mathcal{L}}} \widehat{b}_j \cdot \mu_j = \sum_{j\in\widehat{\mathcal{L}}\cap\mathcal{L}} \widehat{b}_j \cdot \mu_j + \sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}} \widehat{b}_j \cdot \mu_j. \tag{41}$$

To upper bound $\widehat{t}$ we will separately upper bound the two terms on the right hand side of Equation 41. By our assumption that event E_3 does not occur, we have that

$$\begin{aligned}
\sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}} \widehat{b}_j \cdot \mu_j &\leq \sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}} (1+\beta)\cdot\left(\frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j} + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|\right)\cdot\mu_j \\
&\leq \sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}} (1+\beta)\frac{\beta\overline{\mathcal{T}}}{(k+1)} + \sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}} (1+\beta)\beta(|\widetilde{B}_j| + |\widetilde{B}_{j-1}|)\cdot\mu_j \\
&\leq 2\beta\overline{\mathcal{T}} + \sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}} 2\beta(|\widetilde{B}_j| + |\widetilde{B}_{j-1}|)\cdot\mu_j\ ,
\end{aligned} \tag{42}$$

and that

$$\begin{aligned}
\sum_{j\in\widehat{\mathcal{L}}\cap\mathcal{L}} \widehat{b}_j \cdot \mu_j &\leq \sum_{j\in\widehat{\mathcal{L}}\cap\mathcal{L}} (1+\beta)|B''_j| \\
&\leq \sum_{j\in\widehat{\mathcal{L}}\cap\mathcal{L}} (1+\beta)(|B'_j| + \beta|\widetilde{B}_j| + \beta|\widetilde{B}_{j-1}|)\cdot\mu_j \\
&\leq \sum_{j\in\widehat{\mathcal{L}}\cap\mathcal{L}} (1+\beta)|B'_j|\mu_j + \sum_{j\in\widehat{\mathcal{L}}\cap\mathcal{L}} 2\beta(|\widetilde{B}_j| + |\widetilde{B}_{j-1}|)\cdot\mu_j\ ,
\end{aligned} \tag{43}$$

where we used the fact that if event E_1 does not occur, then corollary 3.3.1 holds, implying that for every $j \in [k]$, $B_{I_j} \leq \beta|\widetilde{B}_j|$. Recall that by the definition of the new buckets, for every $j \in [k]$, and every $v \in B'_j$, it holds that $\mu_{j-1} \leq \Delta(v) \leq \mu_j$ and that $\mu_j \leq (1+\beta)^2 \cdot \mu_{j-1}$. Therefore, for every $j \in [k]$, and every $v \in B'_j$, it holds that $\mu_j \leq (1+\beta)^2 \cdot \Delta(v)$, and it follows that $|B'_j|\mu_j \leq (1+\beta)^2|T(B'_j)|$. Also recall that for every $j \in [k]$, it holds that $\mu_j \leq (1+\beta)^j$. The above, together with Equation (42) and Equation (43), implies:

$$\begin{aligned}
\widehat{t} &\leq \sum_{j \in \widehat{\mathcal{L}} \cap \mathcal{L}} (1+\beta)|B'_j|\mu_j + \sum_{j \in \widehat{\mathcal{L}}} 2\beta(|\widetilde{B}_j| + |\widetilde{B}_{j-1}|)\mu_j + 2\beta \cdot \overline{\mathcal{T}} \\
&\leq \sum_{j \in \mathcal{L}} (1+\beta)|B'_j|\mu_j + \sum_{j \in [k']} 2\beta \cdot |\widetilde{B}_j|(1+\beta)^j + \sum_{j \in [k']} 2\beta|\widetilde{B}_{j-1}| \cdot (1+\beta) \cdot (1+\beta)^{j-1} + 2\beta\overline{\mathcal{T}} \\
&\leq (1+\beta)^3 \cdot |\mathcal{T}(B'_{\mathcal{L}})| + 2\beta(1+\beta) \cdot |\mathcal{T}(G)| + 2\beta(1+\beta)^2 \cdot |\mathcal{T}(G)| + 2\beta \cdot |\mathcal{T}(G)| \\
&\leq |\mathcal{T}(B'_{\mathcal{L}})| + 15\beta|\mathcal{T}(G)|. \qquad (44)
\end{aligned}$$

Recall that by Claim 3.3.12,

$$\frac{1}{3}(1-30\beta)|\mathcal{T}(G)| \leq |\mathcal{T}(B'_{\mathcal{L}})|.$$

Therefore,

$$|\mathcal{T}(G)| \leq 3(1+30\beta)|\mathcal{T}(B'_{\mathcal{L}})|.$$

Plugging that into Equation (44), we have that

$$\widehat{t} \leq (1+50\beta)|\mathcal{T}(B'_{\mathcal{L}})|.$$

We now turn to lower bounding $\widehat{t}$. By the assumption that $\frac{1}{3}$-Approx-Triangles estimates the buckets' sizes correctly, Corollary 3.3.16 holds, and it follows that $\widehat{\mathcal{L}} \supseteq \mathcal{L}$. Therefore,

$$\widehat{t} = \sum_{j \in \widehat{\mathcal{L}}} \widehat{b}_j \cdot \mu_j \geq \sum_{j \in \mathcal{L}} (1-\beta)|B'_j| \cdot \mu_j \geq (1-\beta)|\mathcal{T}(B'_{\mathcal{L}})| \,. \qquad (45)$$

The Claim follows from Equations (44) and (45). ■

Theorem 3.2 follows directly from Claims 3.3.18 and Corollary 3.3.14.

3.4 A $(1 \pm \epsilon)$-approximation algorithm

We now present a modified algorithm that computes a $(1 \pm \epsilon)$-approximation of the number of triangles, with probability at least $1 - \frac{1}{\text{polylog n}}$. We start by providing the high-level idea of the modification. In what follows we say that a labeled triangle is *rooted at a bucket*, or more generally, that it is *rooted at a set of vertices* B, if it is rooted at some vertex $v \in B$. Roughly speaking, the $\frac{1}{3}$-approximation algorithm computes an estimate of $3|\text{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}| + 2|\text{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}| + |\text{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}|$ (and divides it by 3). The source of the different factors of 3, 2 and 1 is that the algorithm only estimates large significant buckets, which accounts for 3 "copies" of each triangle in $\text{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}$ (one for each endpoint), 2 "copies" of each triangle in $\text{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}$ and one "copy" for each triangle in $\text{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}$.

The $(1 \pm \epsilon)$-approximation algorithm aims at estimating $|\text{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}|$ and $|\text{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}|$. If we could sample triangles uniformly and determine for a given triangle to which subset it belongs, then we would obtain such estimates.

Observe that for each triangle (v,u,w) in $\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}$ or $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}$ there is at least one vertex, say v, that belongs to a large significant bucket. Recall that we are able to obtain a good estimate, $\hat{t}$, for the number of labeled triangles $(v,u,w)_v$ that are rooted at vertices v belonging to large significant buckets. Let α_1 denote the fraction among these labeled triangles in which the two other vertices, u and w, both belong to small significant buckets, and let α_2 denote the fraction in which only one of them belongs to a small significant bucket. If we could obtain good estimates for α_1 and for α_2, then, together with $\hat{t}$, we would get good estimates for $|\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}|$ and $|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}|$, respectively. It hence remains to explain how to obtain such estimates.

Suppose we could (efficiently) sample labeled triangles $(v,u,w)_v$ uniformly among all labeled triangles rooted at vertices v belonging to large buckets and that we could (efficiently) determine for any vertex of our choice whether it belongs to a small significant bucket or a large one. Then we could easily obtain good estimates for α_1 and for α_2. While we cannot perform these tasks precisely, we can do so approximately, which is sufficient for our needs. Specifically, we can do the following.

1. Using our estimates, $\hat{b}_j$, for the sizes of the large significant buckets, we can select a large significant bucket with a probability that is roughly proportional to the number of labeled triangles rooted at the bucket. Namely, we select a bucket index j with probability $\hat{b}_j\mu_j/\hat{t}$ where $\hat{t}$ is the sum, over all large buckets, of $\hat{b}_j\mu_j$.

2. We can select a vertex roughly uniformly in a large significant bucket B_j. This is done by selecting a sufficiently large sample of vertices and for each sampled vertex v calling Approx-Triangles-Degree(v,j) until we get an estimate $\widehat{\Delta}(v)$ indicating that $v \in B_j$.

3. We can select a triangle $(v,u,w)_v$ rooted at v roughly uniformly. This is done by slightly modifying the procedure Approx-Triangles-Degree and the procedures that it calls so that instead of returning an estimate of $\Delta(v)$ it returns a triangle rooted at v.

4. Given a triangle $(v,u,w)_v$, we can determine (approximately) for u (similarly, for w) whether it belongs to a small significant bucket by calling a modified version of Approx-Triangles-Degree.

We now turn to provide more precise details of the algorithm and its analysis.

3.4.1 The compensation idea

Definition 3.4.1 *Let $\widehat{\mathcal{L}}$ be as defined in Step (12) of $\frac{1}{3}$-Approx-Triangles, and let $\widehat{\mathcal{S}}$ be the set of indices of significant buckets not in $\widehat{\mathcal{L}}$. That is, $\widehat{\mathcal{S}} = \left\{ j \in [k] \setminus \widehat{\mathcal{L}} \ : \ B'_j \geq \frac{\beta\overline{T}}{(k+1)\cdot\mu_j} \right\}$.*

Definition 3.4.2 *Let $\mathcal{L}$ be as defined in Definition 3.3.12. Define $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ to be*

$$\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = \frac{|Tr_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}|}{|\mathcal{T}(B'_{\mathcal{L}})|} \quad \text{and} \quad \alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} = \frac{|Tr_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}|}{|\mathcal{T}(B'_{\mathcal{L}})|}.$$

Claim 3.4.1 *For $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ as defined in Definition 3.4.2,*

$$(1-10\beta)|\mathcal{T}(G)| \ \leq \ |\mathcal{T}(B'_{\mathcal{L}})|(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \ \leq \ |\mathcal{T}(G)|. \tag{46}$$

Proof: By the definition of $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$,

$$\begin{aligned}
&|\mathcal{T}(B'_{\mathcal{L}})|(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \\
&\quad = 3|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}| + 2|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}| + |\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}| + 2|\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}| + |\mathrm{Tr}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}| \\
&\quad = 3|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}| + 3|\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}| + 3|\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}| \\
&\quad\quad - 2|\mathrm{Tr}_{\mathcal{L},\mathcal{S}\setminus\widehat{\mathcal{S}},\mathcal{S}\setminus\widehat{\mathcal{S}}}| - 2|\mathrm{Tr}_{\mathcal{L},\mathcal{S}\setminus\widehat{\mathcal{S}},\mathcal{S}\setminus\widehat{\mathcal{S}}}| - |\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}\setminus\widehat{\mathcal{S}}}|. \qquad (47)
\end{aligned}$$

Observe that for every index $j \in \mathcal{S} \setminus \widehat{\mathcal{S}}$, it holds that B'_j is insignificant. This is true, since for every significant bucket B'_j there are two possibilities. If $j \leq k'$, then j is in $\mathcal{L}$, and therefore, $j \notin S$. If $j > k'$ then $j \notin \widehat{\mathcal{L}}$, and therefore $j \in \widehat{\mathcal{S}}$. It follows that the triangles in the sets $\mathrm{Tr}_{\widehat{\mathcal{L}},\mathcal{S}\setminus\widehat{\mathcal{S}},\mathcal{S}\setminus\widehat{\mathcal{S}}}$, $\mathrm{Tr}_{\widehat{\mathcal{L}},\mathcal{S}\setminus\widehat{\mathcal{S}},\mathcal{S}\setminus\widehat{\mathcal{S}}}$ and $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}\setminus\widehat{\mathcal{S}}}$ are contained in Tr_I. Further observe that the only triangles not accounted for in the sum in Equation (47) are triangles with at least one endpoint in an insignificant bucket or in a safety interval. It follows that

$$|\mathcal{T}(B'_{\mathcal{L}})|(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \geq |\mathcal{T}(G)| - 3|\mathrm{Tr}_I| \geq (1-10\beta)|\mathcal{T}(G)|, \qquad (48)$$

where the last inequality follows from Equation (35) in the proof of Claim 3.3.12. Also, since the sets $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{L}}$, $\mathrm{Tr}_{\mathcal{L},\mathcal{L},\mathcal{S}}$ and $\mathrm{Tr}_{\mathcal{L},\mathcal{S},\mathcal{S}}$ in Equation (47) are disjoint,

$$|\mathcal{T}(B'_{\mathcal{L}})|(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \leq |\mathcal{T}(G)|. \qquad (49)$$

The claim follows from Equations (48) and (49). ■

3.4.2 Sampling a random triangle and determining its type

We modify the procedure Approx-Triangles-Degree so that when invoked with a vertex v and an index j, the procedure computes $\widehat{\Delta}(v)$, and if $\widehat{\Delta}(v) \in (\mu_{j-1}, \mu_j]$, then the procedure returns a roughly uniform triangle rooted at v. We do so by using slightly altered versions of the procedures Approx-Triangles-Of-Low-Deg-Vertices, Approx-High-Triangles and Approx-Crossing-Triangles such that each procedure, when invoked with a vertex v, returns both the approximated triangles-degree and the first triangle seen during the run of the procedure (or an empty set if no such triangle was seen).

Definition 3.4.3 *We say that the modified versions Approx-Triangles-Of-Low-Deg-Vertices', Approx-High-Triangles' and Approx-Crossing-Triangles' answer correctly if the following holds. First, the procedures return $\widehat{\Delta}$ for which the statements in Definitions 3.3.2, 3.3.4 and 3.3.5, respectively. Second, if the procedures return the first triangles triangle "seen" during the run of the procedure or an empty set if no triangle was seen.*

Claim 3.4.2 *Claims 3.3.4, 3.3.5, 3.3.6, 3.3.7, 3.3.8, and 3.3.9 hold for the modified versions Approx-Triangles-Of-Low-Deg-Vertices', Approx-High-Triangles' and Approx-Crossing-Triangles' and for the new definition of answering correctly as defined in Definition 3.4.3.*

Proof: The claim follows directly from the definitions of the new procedures and from the proofs of the listed claims. ■

Procedure 7 Sample-Random-Triangle(v, j)

Let $\delta \leftarrow \frac{1}{7} \cdot \frac{\gamma}{\log n} \cdot \frac{\beta}{1+\beta}$.
Query v's degree $d(v)$.
If $d(v) \leq \sqrt{m}$ **then**
    $\langle \widehat{\Delta},\ (v,u,w)_v \rangle \leftarrow$ Approx-Triangles-Of-Low-Deg-Vertices'$(v, j, d(v), \delta)$.
    If $\widehat{\Delta} \in (\mu_{j-1}, \mu_j]$ **then**
        Return $(v,u,w)_v$.
Else
    $\langle \widehat{\Delta}_{hi},\ (v,u,w)_v \rangle \leftarrow$ Approx-High-Triangles'$(v, j, d(v), \delta)$.
    $\langle \widehat{\Delta}_{cr},\ (v,u',w')_v \rangle \leftarrow$ Approx-Crossing-Triangles'$(v, j, d(v), \delta)$.
    $\widehat{\Delta} \leftarrow \widehat{\Delta}_{hi} + \widehat{\Delta}_{cr}$.
    If $\widehat{\Delta} \in (\mu_{j-1}, \mu_j]$ **then**
        Return $\begin{cases} (v,u,w)_v & \text{with probability } \widehat{\Delta}_{hi}/\widehat{\Delta} \\ (v,u',w')_v & \text{with probability } \widehat{\Delta}_{cr}/\widehat{\Delta} \end{cases}$.
Return $\emptyset$.

Claim 3.4.3 *Assume that, if invoked, the procedures Approx-Triangles-Of-Low-Deg-Vertices', Approx-High-Triangles' and Approx-Crossing-Triangle' answer correctly as defined in Definition 3.4.3. For every vertex v and for every $j \in [k']$, Sample-Random-Triangle(v, j) satisfies the following:*

1. *If $v \in B'_j$, then the procedure returns a triangle, and for all but at most a δ fraction of the triangles rooted at v, the probability that each of them is returned by Sample-Random-Triangle(v, j) is in $\left(\frac{1-5\delta}{|\Delta(v)|}, \frac{1+5\delta}{|\Delta(v)|}\right)$.*
2. *If $v \notin B''_j$ then Sample-Random-Triangle(v, j) returns an empty set.*
3. *If $v \in B''_j \setminus B'_j$, then Sample-Random-Triangle(v, j) either returns an empty set or a roughly uniform triangle rooted in v as described in Item (1).*

Proof: If $v \notin B''_j$, then by Claim 3.4.2 and by Corollary 3.3.11, $\widehat{\Delta} \notin (\mu_{j-1}, \mu_j]$, and the procedure will return an empty set. This proves the second item of the claim.

To prove the first item of the claim, consider a vertex $v \in B'_j$. First assume that the vertex v is such that $d(v) \leq \sqrt{m}$. By the assumption that Approx-Triangles-Of-Low-Deg-Vertices' answers correctly it follows that it returns $\widehat{\Delta}$ such that $\widehat{\Delta} \in (\mu_{j-1}, \mu_j]$, implying that it also returns a triangle and not an empty set (if $\widehat{\Delta} \neq 0$ then at least one triangle was seen during the run of Approx-Triangles-Of-Low-Deg-Vertices'). All the triangles rooted at v have exactly the same probability of being the first one seen during the run of Approx-Triangles-Of-Low-Deg-Vertices'. Hence, each triangle rooted in v is returned with probability $\frac{1}{|\Delta(v)|}$.

Now consider a vertex v such that $d(v) > \sqrt{m}$. As before, if the procedure Approx-Crossing-Triangles' returns $\widehat{\Delta}_{cr} > 0$, then it also returns a uniform triangle in the set $\mathrm{Tr}_{cr}(v)$. Similarly,

if the procedure Approx-High-Triangles' returns $\widehat{\Delta}_{hi} > 0$, then it also returns a uniform triangle among the set $\mathrm{Tr}_{hi}(v)$. Therefore, for each triangle in $\mathrm{Tr}_{hi}(v)$ and for each triangle in $\mathrm{Tr}_{cr}(v)$, the probability they it is returned is

$$\frac{\widehat{\Delta}_{hi}}{\widehat{\Delta}} \cdot \frac{1}{\Delta_{hi}(v)} \quad \text{and} \quad \frac{\widehat{\Delta}_{cr}}{\widehat{\Delta}} \cdot \frac{1}{\Delta_{cr}(v)}$$

respectively. If both $\Delta_{cr}(v)$ and $\Delta_{hi}(v)$ are at least $\delta \cdot \mu_{j-1}$, then by the assumption that Approx-Crossing-Triangles' and Approx-High-Triangles' answer correctly, it holds that $(1-\delta)\Delta_{hi}(v) \leq \widehat{\Delta}_{hi} \leq (1+\delta)\Delta_{hi}(v)$ and that $(1-\delta)\Delta_{cr}(v) \leq \widehat{\Delta}_{hi} \leq (1+\delta)\Delta_{cr}(v)$. The above, together with Lemma 3.3.10, implies that the probability of a specific triangle in $\mathrm{Tr}_{hi}(v)$ to be returned is:

$$\frac{(1-\delta)\Delta_{hi}(v)}{\Delta_{hi}(v)} \cdot \frac{1}{(1+3\delta)\Delta(v)} \leq \frac{\widehat{\Delta}_{hi}}{\widehat{\Delta}} \cdot \frac{1}{\Delta_{hi}(v)} \leq \frac{(1+\delta)\Delta_{hi}(v)}{\Delta_{hi}(v)} \cdot \frac{1}{(1-3\delta)\Delta(v)},$$

implying that for every triangle $(v,u,w) \in \mathrm{Tr}_{hi}(v)$,

$$\frac{1-5\delta}{\Delta(v)} \leq \Pr[(v,u,w)_v \text{ is returned}] \leq \frac{1+5\delta}{\Delta(v)}. \tag{50}$$

Similarly for every triangle $(v,u',w') \in \mathrm{Tr}_{cr}(v)$,

$$\frac{1-5\delta}{\Delta(v)} \leq \Pr[(v,u',w')_v \text{ is returned}] \leq \frac{1+5\delta}{\Delta(v)}. \tag{51}$$

Therefore, every triangle in $\mathrm{Tr}(v)$ is returned with probability $(1 \pm 5\delta)\frac{1}{\Delta(v)}$ as required.

Since $\widehat{\Delta}(v) \in (\mu_{j-1}, \mu_j]$ it could be that either neither or one of $\Delta_{cr}(v)$ or $\Delta_{hi}(v)$ is smaller than $\delta \cdot \mu_{j-1}$, but not both. Assume without loss of generality that $\Delta_{cr}(v) < \delta \cdot \mu_{j-1}$. The probability of each triangle in $\mathrm{Tr}_{hi}(v)$ to be returned is as in Equation (50). Therefore, at least $(1-\delta)$ fraction of the triangles rooted at v are (each) returned with probability $(1 \pm 5\delta)\frac{1}{\Delta(v)}$. This concludes the proof of the first item of the claim.

For a vertex $v \in B''_j \setminus B'_j$ it could be that either $\widehat{\Delta}(v) \notin (\mu_{j-1}, \mu_j]$ and the procedure returns an empty set, or that $\widehat{\Delta}(v) \in (\mu_{j-1}, \mu_j]$, in which case the first item holds. Therefore the third item of the Claim holds, and the proof is complete. ■

Claim 3.4.4 *For a vertex v let $Tr'(v) \subseteq Tr(v)$ be any subset of the triangles rooted at v. Assume that, if invoked, the procedure Approx-High-Triangles' and Approx-Crossing-Triangle' answer correctly as defined in Definition 3.4.3. If Sample-Random-Triangle returns a triangle rooted at v, then the probability it returns a triangle from $Tr'(v)$ is in*

$$\left[\frac{(1-5\delta)(|Tr'(v)| - \delta \cdot \mu_{j-1})}{\Delta(v)}, \frac{(1+5\delta)|Tr'(v)|}{\Delta(v)}\right].$$

Proof: First consider a low-degree vertex v. By the definition of the procedure Approx-Triangles-Of-Low-Deg-Vertices', every triangle in $\mathrm{Tr}(v)$ has the same probability to be the first one seen during the run of the procedure. Therefore, the procedure returns a triangle from $\mathrm{Tr}'(v)$ with probability

$\frac{|\mathrm{Tr}'(v)|}{\Delta(v)}$, and the claim follows. Now consider a high-degree vertex v. The probability that a triangle from $\mathrm{Tr}'(v)$ is returned is:

$$\sum_{(v,u,w)_v \in \mathrm{Tr}'(v)} \Pr[(v,u,w)_v \text{ is returned}] =$$
$$\sum_{\substack{(v,u,w)_v \in \\ \mathrm{Tr}'(v) \cap \mathrm{Tr}_{hi}(v)}} \Pr[(v,u,w)_v \text{ is returned}] \quad + \quad \sum_{\substack{(v,u,w)_v \in \\ \mathrm{Tr}'(v) \cap \mathrm{Tr}_{cr}(v)}} \Pr[(v,u,w)_v \text{ is returned}].$$

If both $\Delta_{cr}(v)$ and $\Delta_{hi}(v)$ are at least $\delta \cdot \mu_{j-1}$, then Equations (50) and (51) from the proof of Claim 3.4.3 hold, and the claim follows. Observe that it is not possible that both $\Delta_{cr}(v)$ and $\Delta_{hi}(v)$ are smaller than $\delta \cdot \mu_{j-1}$ and that Sample-Random-Triangle returns a triangle, since the procedure only returns a triangle if $\widehat{\Delta} \in (\mu_{j-1}, \mu_j]$. Therefore, assume without loss of generality that $\Delta_{cr}(v) < \delta \cdot \mu_{j-1}$. In such a case

$$|\mathrm{Tr}'(v) \cap \mathrm{Tr}_{hi}(v)| = |\mathrm{Tr}'(v)| - |\mathrm{Tr}'(v) \cap \mathrm{Tr}_{cr}(v)| \geq |\mathrm{Tr}'(v)| - \Delta_{cr}(v) \geq |\mathrm{Tr}'(v)| - \delta \cdot \mu_{j-1} \,.$$

Since $\Delta_{hi}(v) \geq \delta \cdot \mu_{j-1}$, Equation (50) from the proof of Claim 3.4.3 holds. It follows that

$$\sum_{(v,u,w)_v \in \mathrm{Tr}'(v)} \Pr[(v,u,w)_v \text{ is returned}] \geq (1-5\delta) \cdot \frac{|\mathrm{Tr}'(v) \cap \mathrm{Tr}_{hi}(v)|}{\Delta(v)}$$
$$\geq (1-5\delta) \cdot \frac{|\mathrm{Tr}'(v)| - \delta\mu_{j-1}}{\Delta(v)} \,, \tag{52}$$

and

$$\sum_{(v,u,w)_v \in \mathrm{Tr}'(v)} \Pr[(v,u,w)_v \text{ is returned}] \leq (1+5\delta) \cdot \frac{|\mathrm{Tr}'(v) \cap \mathrm{Tr}_{hi}(v)|}{\Delta(v)}$$
$$\leq (1+5\delta) \cdot \frac{|\mathrm{Tr}'(v)|}{\Delta(v)} \,. \tag{53}$$

The claim follows from Equations (52) and (53). ■

We now give a procedure for estimating whether a given vertex belongs to a small significant bucket:

Procedure 8 Is-Small-And-Significant(u)

1: Let $\widehat{\Delta}(u) \leftarrow$ Approx-Triangles-Degree$(u, \frac{1}{2}(\beta\overline{\mathcal{T}})^{2/3})$.
2: Let ℓ be such that $\widehat{\Delta}(u) \in (\mu_{\ell-1}, \mu_\ell]$.
3: **If** $\widehat{\Delta}(u) < (1-\delta)(\beta\overline{\mathcal{T}})^{2/3}$ **or** $\ell \in \widehat{\mathcal{L}}$ **then**
4: **Return** 0.
5: **Else**
6: **Return** 1.

Claim 3.4.5 *Assume that Approx-Triangles-Degree answers correctly on the sampled vertex in Step (1) of Is-Small-And-Significant, and that Item (1) in Definition 3.3.13 holds. It holds that:*

1. *If* $u \in B'_{\widehat{\mathcal{S}}}$, *then* Is-Small-And-Significant$(u) = 1$.

2. *If* $u \in B'_{\widehat{\mathcal{L}}}$, *then* Is-Small-And-Significant$(u) = 0$.

Proof: First consider a vertex $u \in B'_{\widehat{\mathcal{S}}}$ and let B'_j be the bucket it belongs to. By the definition of $\widehat{\mathcal{S}}$, $|B'_j| \geq \frac{\beta\overline{\mathcal{T}}}{(k+1)\mu_j}$, implying that $\mu_j \geq \frac{\beta\overline{\mathcal{T}}}{(k+1)|B'_j|}$. By the assumption that Item (1) in Definition 3.3.13 holds, Corollary 3.3.16 holds, and therefore $\mathcal{L} \subseteq \widehat{\mathcal{L}}$. It follows that for every $j \notin \widehat{\mathcal{L}}$, B'_j is small i.e., $|B'_j| < \frac{(\beta\overline{\mathcal{T}})^{1/3}}{(k+1)}$. This implies that $\mu_j \geq (\beta\overline{\mathcal{T}})^{2/3}$ and that $\mu_{j-1} \geq \mu_j(1+\beta)^2 \geq \frac{1}{2}(\beta\overline{\mathcal{T}})^{2/3}$. By the assumptions that u belongs to a strict bucket and that Approx-Triangles-Degree answers correctly and by Corollary 3.3.11, u will be assigned to its correct bucket. That is, ℓ as computed in Step (2) is such that $u \in B'_\ell$. Therefore, $\ell \notin \widehat{\mathcal{L}}$, implying that Is-Small-And-Significant$(u) = 1$, and the proof of the first item is complete.

Now consider a vertex u such that $u \in B'_j$ and $j \in \mathcal{L}'$. If $\Delta(u) < \frac{1}{2}(\beta\overline{\mathcal{T}})^{2/3}$ then by Lemma 3.3.10 and by the assumption that Approx-Triangles-Degree answers correctly, $\widehat{\Delta}(u) < (1+\delta) \cdot \frac{1}{2}(\beta\overline{\mathcal{T}})^{2/3} < (1-\delta)(\beta\overline{\mathcal{T}})^{2/3}$ and the procedure will return 0. Otherwise, $\Delta(u) \geq \frac{1}{2}(\beta\overline{\mathcal{T}})^{2/3}$, so that by the assumption that Approx-Triangles-Degree answers correctly, and by Corollary 3.3.11, u will be assigned to its correct bucket B'_j. By the assumption that $j \in \widehat{\mathcal{L}}$ it follows that $\ell \in \widehat{\mathcal{L}}$ and that the procedure will return 0, as required. ■

3.4.3 The algorithm for $(1 \pm \epsilon)$-approximation

Algorithm 9 Approx-Triangles-With-Advice $(G, \overline{\Delta}, \overline{m}, \epsilon)$

1: Invoke $\frac{1}{3}$-Approx-Triangles$(G, 3\overline{\Delta}, \overline{m}, \epsilon)$, and let $\{\mu_0, \ldots, \mu_k\}$, $\widehat{b}_j$ and $\widehat{t}$ be as defined in the algorithm.

2: **For** $i = 1, \ldots, r = \frac{20 \log n}{\beta^3}$ **do**

3: Pick an index j with probability $\frac{\widehat{b}_j \mu_{j-1}}{\widehat{t}}$. Denote the selected index j_i.

4: **Repeat** at most $s_{j_i} = \frac{n \cdot \mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n \cdot (k+1)}{\beta^4}$ times

Sample a vertex v until Sample-Random-Triangle(v, j_i) returns a triangle $(v, u, w)_v$.

5: Let $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = \begin{cases} 0 & \text{No triangle was sampled in Step (4)} \\ 1 & \text{A triangle } (v,u,w)_v \text{ was sampled and both Is-Small-And-Significant}(u) = 1 \\ & \text{and Is-Small-And-Significant}(w) = 1 \\ 0 & \text{otherwise} \end{cases}$.

6: Let $\chi^i_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} = \begin{cases} 0 & \text{No triangle was sampled in Step (4)} \\ 1 & \text{if either Is-Small-And-Significant}(u) = 1 \text{ or Is-Small-And-Significant}(w) = 1 \\ & \text{but not both} \\ 0 & \text{otherwise} \end{cases}$.

7: $\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leftarrow \frac{1}{r} \cdot \sum_i \chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$.

8: $\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \leftarrow \frac{1}{r} \cdot \sum_i \chi^i_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$.

9: **Return** $\widehat{\Delta} = \frac{1}{3} \cdot \widehat{t} \cdot (1 + 2\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + \frac{1}{2}\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}})$

Definition 3.4.4 *Consider the i^{th} iteration of the loop in Step (2) of Approx-Triangles-With-Advice. For every i, if the chosen index in Step (3), j_i, is such that $j_i \in \mathcal{L}$ and a triangle $(v,u,w)_v$ is sampled in Step (4) then we say that the i^{th} iteration was successful, and if $j_i \in \mathcal{L}$ and no triangle was sampled in Step (4) we say that the iteration was unsuccessful.*

Claim 3.4.6 *For every iteration of the loop in Step (2) of the algorithm Approx-Triangles-With-Advice, the sampling process in Steps (3) and (3) is successful with probability at least $1 - \frac{1}{n^2}$.*

Proof: By the first item in Claim 3.4.3, if Sample-Random-Triangle is invoked with a vertex v and an index j_i such that $v \in B'_{j_i}$, then a triangle is returned. Therefore, the probability that a triangle is returned in any iteration of the loop in Step (4) is at least $\frac{|B'_{j_i}|}{n}$. For indices j_i such that j_i in $\mathcal{L}$, it holds that $|B'_{j_i}| \geq \frac{\beta\overline{\mathcal{T}}}{(k+1)\cdot\mu_j}$. Therefore the probability that a triangle is not returned in s_{j_i} tries is at most

$$\left(1 - \frac{\beta\overline{\mathcal{T}}}{n \cdot (k+1) \cdot \mu_j}\right)^{\frac{n\cdot\mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n \cdot (k+1)}{\beta^4}} < \exp(-10 \cdot \frac{\log n}{\beta^3}) < \frac{1}{\beta^3} \cdot \frac{1}{n^3} .$$

By applying the union bound over all the iterations we get that the probability that any of the iterations fails is at most $\frac{1}{n^2}$. ■

Definition 3.4.5 *We say that $\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ is a good estimation of $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ if $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \geq 71\beta$ and $(1-\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}-70\beta) \leq \widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leq (1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+70\beta)$ or if $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} < 71\beta$ and $\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leq 150\beta$. Similarly, we say that $\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ is a good estimation of $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ if $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \geq 71\beta$ and $(1-\beta)(\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} - 70\beta) \leq \widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \leq (1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta)$ or if $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} < 71\beta$ and $\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \leq 150\beta$.*

Lemma 3.4.7 *Assume that Create-Random-Thresholds returns a good selection and that Approx-Triangles-Degree answers correctly on all the sampled vertices. Further assume that Item (1) and Item (2) in Definition 3.3.13 hold for every $j \in [k']$ and that every iteration of the loop in Step (2) of Approx-Triangles-With-Advice is successful. Let $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ be as defined in Definition 3.4.2, and let $\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ be as defined in Approx-Triangles-With-Advice. With probability at least $1-1/n^3$,*

1. *$\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ is a good estimation of $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$.*
2. *$\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ is a good estimation of $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$.*

Before proving the lemma we prove the following two claims.

Claim 3.4.8 *Let $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\chi^i_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ be as defined in Steps (5) and (6) in Approx-Triangles-With-Advice. If the assumption in Lemma 3.4.7 hold, then*

$$\Pr[\chi^{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)=1] \geq \alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 70\beta,$$

and

$$\Pr[\chi^{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}(v)=1] \geq \alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} - 70\beta.$$

Proof: We start by lower bounding the probability $\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}] = 1$. For a vertex $v \in B_{\mathcal{L}}$, let $\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)$ be the set of triangles $(v,u,w)_v$ rooted at v such that both u and w are in $B'_{\widehat{\mathcal{S}}}$, and let $\Delta_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v) = |\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)|$. Recall that $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = 1$ if a triangle $(v,u,w)_v$ was sampled in Step (4) for which Is-Small-And-Significant$(u) = 1$ and Is-Small-And-Significant$(w) = 1$. Therefore, by Item (1) in Claim 3.4.5 and Claim 3.4.6, for every vertex v such that $v \in B'_j$ for some $j \in \widehat{\mathcal{L}}$ if the sampled triangle is in $\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)$ then $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = 1$. By Claim 3.4.4 the probability that the sampled triangle in Step (4) is from $\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)$ is at least $\frac{(1-5\delta)\cdot(\Delta_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)-\delta\cdot\mu_{j-1})}{\Delta(v)}$. Also note that, under the assumption that Approx-Triangles-Degree answers correctly on all the sampled vertices, for every index $j \in \mathcal{L}$ its probability to be the chosen index in Step (3) is $\frac{\widehat{b}_j\cdot\mu_j}{\widehat{t}}$, and for every vertex v in B'_j its probability to be the vertex for which a triangle is sampled in Step (4) is at least $\frac{1}{|B''_j|}$. Combining the above we get that:

$$\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}=1] \geq \sum_{j\in\mathcal{L}} \frac{\widehat{b}_j\cdot\mu_{j-1}}{\widehat{t}} \cdot \sum_{v\in B'_j} \frac{1}{|B''_j|} \cdot \frac{(1-5\delta)(\Delta_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)-\delta\mu_{j-1})}{\Delta(v)} \,. \tag{54}$$

For every $v \in B'_j$, it holds that $\mu_{j-1} \leq \Delta(v) \leq \mu_j$, implying that

$$\frac{\mu_{j-1}}{\Delta(v)} \geq \frac{\mu_{j-1}}{\mu_j} \geq \frac{1}{(1+\beta)^2} > 1 - 3\beta \,. \tag{55}$$

By Corollary 3.3.17 for every $j \in \widehat{\mathcal{L}}$,

$$\widehat{b}_j \geq (1-\beta)|B'_j|.$$

Therefore,

$$\frac{\widehat{b}_j}{|B''_j|} \geq \frac{(1-\beta)|B'_j|}{|B''_j|} \geq \frac{(1-\beta)(|B''_j| - \beta|\widetilde{B}_j| - \beta|\widetilde{B}_{j-1}|)}{|B''_j|} \geq 1 - \beta - 2\beta(1-\beta) > 1 - 3\beta. \tag{56}$$

Clearly, $\widehat{\mathcal{L}} \subset [k]$, and recall that according to Corollary 3.3.16, $\widehat{\mathcal{L}} \supseteq \mathcal{L}$. Therefore, by plugging Equations (56) and (55) into Equation (54) we get

$$\begin{aligned}
\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = 1] &\geq (1-3\beta)^2(1-5\delta) \cdot \left(\sum_{j \in \widehat{\mathcal{L}}} \sum_{v \in B'_j} \frac{\Delta_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)}{\widehat{t}} - \sum_{j \in \widehat{\mathcal{L}}} \sum_{v \in B'_j} \frac{\delta \cdot \mu_{j-1}}{\widehat{t}} \right) \\
&\geq (1-8\beta) \sum_{j \in \mathcal{L}} \sum_{v \in B'_j} \frac{\Delta_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)}{\widehat{t}} - (1-8\beta) \sum_{j \in [k]} \sum_{v \in B'_j} \frac{\delta \cdot \mu_{j-1}}{\widehat{t}} \\
&\geq (1-8\beta) \cdot \frac{|\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}|}{\widehat{t}} - 2\frac{\delta|\mathcal{T}(G)|}{\widehat{t}} \,.
\end{aligned}$$

Recall that from Claim 3.3.18, $\widehat{t} \leq (1+50\beta)|\mathcal{T}(B'_{\mathcal{L}})|$. Also, it follows from Theorem 3.2 that

$$\frac{|\mathcal{T}(G)|}{\widehat{t}} \leq \frac{3}{1-50\beta} \leq 4. \tag{57}$$

Therefore,

$$\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = 1] \geq (1-8\beta)\frac{|\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}|}{(1+50\beta)|\mathcal{T}(B'_{\mathcal{L}})|} - 8\beta \geq (1-60\beta)\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 8\beta \;\geq \alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 70\beta, \tag{58}$$

as claimed.

The proof for the second part of the claim follows in an almost identical manner. ■

Claim 3.4.9 *Let $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\chi^i_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ be as defined in Steps (5) and 6 in Approx-Triangles-With-Advice. If the assumption in Lemma 3.4.7 hold, then*

$$\Pr[\chi^{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v) = 1] \leq \alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta,$$

and

$$\Pr[\chi^{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}(v) = 1] \leq \alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} + 70\beta.$$

Proof: We will only upper bound the probability $\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}] = 1$, aa the proof for upper bounding $\Pr[\chi^{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}(v) = 1]$ is almost identical. For a vertex $v \in B''_{\widehat{\mathcal{L}}}$, denote by $\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)$ the set of triangles $(v,u,w)_v$ rooted at v such that both u and w are in $[k]\setminus\widehat{\mathcal{L}}$. Observe that for any triangle $(v,u,w)_v$ rooted in v that is not in the set $\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)$, either u or w are in $\widehat{\mathcal{L}}$, and therefore, by Item (2) in Claim 3.4.5, if such a triangle is selected in Step (3) of the algorithm, then $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ is set to 0.

For every index j chosen in Step (3) of the algorithm, the probability that $\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = 1$ (conditioned on the choice of j) is upper bounded by the maximum, over all subsets $\overline{B}_j$ such that $B'_j \subseteq \overline{B}_j \subseteq B''_j$ of the following expression:

$$\begin{aligned}
&\sum_{v\in\overline{B}_j}\frac{1}{|\overline{B}_j|}\cdot\min\left\{1,\frac{(1+5\delta)|\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)|}{\Delta(v)}\right\}\\
&\quad\leq\quad\frac{1}{|\overline{B}_j|}\left(\sum_{v\in\mathrm{B}'_j}\min\left\{1,\frac{(1+5\delta)|\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)|}{\Delta(v)}\right\}+|B_{I_j}\cap\overline{B}_j|\right)\ . \qquad (59)
\end{aligned}$$

The right-hand side of Equation (59) is maximized when $\overline{B}_j = B''_j$. Therefore,

$$\begin{aligned}
\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}=1] &\leq \sum_{j\in\widehat{\mathcal{L}}}\frac{\widehat{b}_j\cdot\mu_{j-1}}{\widehat{t}}\cdot\left(\frac{1}{|\mathrm{B}''_j|}\cdot\sum_{v\in\mathrm{B}'_j}\min\left\{1,\frac{(1+5\delta)|\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)|}{\Delta(v)}\right\}+\frac{|B_{I_j}|}{|B''_j|}\right)\\
&\leq \sum_{j\in\mathcal{L}}\frac{\widehat{b}_j\cdot\mu_{j-1}}{\widehat{t}}\cdot\left(\frac{1}{|\mathrm{B}''_j|}\cdot\sum_{v\in\mathrm{B}'_j}\min\left\{1,\frac{(1+5\delta)|\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)|}{\Delta(v)}\right\}+\frac{|B_{I_j}|}{|B''_j|}\right)\\
&\quad+\sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}}\frac{\widehat{b}_j\cdot\mu_{j-1}}{\widehat{t}}\cdot\left(\frac{|B'_j|}{|\mathrm{B}''_j|}+\frac{|B_{I_j}|}{|B''_j|}\right)\\
&\leq \frac{1}{\widehat{t}}\cdot\sum_{j\in\mathcal{L}}\frac{\widehat{b}_j}{|B''_j|}\cdot\sum_{v\in\mathrm{B}'_j}\frac{\mu_{j-1}}{\Delta(v)}\cdot(1+5\delta)|\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)|\ +\ \frac{1}{\widehat{t}}\cdot\sum_{j\in\mathcal{L}}\frac{\widehat{b}_j}{|B''_j|}\cdot|B_{I_j}|\cdot\mu_{j-1}\\
&\quad+\frac{1}{\widehat{t}}\sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}}\widehat{b}_j\cdot\mu_{j-1}\ . \qquad (60)
\end{aligned}$$

For every $v\in B''_j$, $\Delta(v)\geq\mu_{j-2}$, implying that

$$\frac{\mu_{j-1}}{\Delta(v)}\leq\frac{\mu_{j-1}}{\mu_{j-2}}\leq(1+\beta)^2\leq 1+3\beta\ . \qquad (61)$$

By the assumption that Item (1) in Definition 3.3.13 holds, for every $j\in\mathcal{L}$, it holds that $\widehat{b}_j\leq(1+\beta)|B''_j|$. We also use the fact that $\Delta(v)\geq\mu_{j-1}$ for every $j\in[k]$ and $v\in B'_j$. Starting with the last term in Equation (60), it follows from Equation (42) in the proof of Claim 3.3.15, that

$$\frac{1}{\widehat{t}}\cdot\sum_{j\in\widehat{\mathcal{L}}\setminus\mathcal{L}}\widehat{b}_j\cdot\mu_{j-1}\leq\frac{6\beta\overline{\mathcal{T}}}{\widehat{t}}\leq\frac{6\beta|\mathcal{T}(G)|}{\widehat{t}}\leq 24\beta\ . \qquad (62)$$

Turning to the second term, it follows from Equation (33) in the proof of Claim 3.3.12 that

$$\frac{1}{\widehat{t}} \cdot \sum_{j \in \mathcal{L}} \frac{\widehat{b}_j}{|B''_j|} \cdot |B_{I_j}| \cdot \mu_{j-1} \leq \frac{1}{\widehat{t}} \sum (1+\beta)|B_{I_j}| \cdot \mu_{j-1} \leq \frac{(1+\beta)\beta|\mathcal{T}(G)|}{\widehat{t}} \leq 8\beta \,. \tag{63}$$

And finally, in order to bound the first term, observe that a triangle $(v,u,w)_v$ is in $\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}} \setminus \mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ if it has at least one endpoint in either $\widehat{\mathcal{L}} \setminus \mathcal{L}$ or in $([k] \setminus \widehat{\mathcal{L}}) \setminus \widehat{\mathcal{S}}$. That is, if it has at least one endpoint in an insignificant bucket or in a safety interval. Hence the first term is upper bounded by

$$\begin{aligned}
&\frac{1}{\widehat{t}} \cdot \sum_{j \in \mathcal{L}} \frac{\widehat{b}_j}{|B''_j|} \cdot \sum_{v \in \mathrm{B}'_j} \frac{\mu_{j-1}}{\Delta(v)} \cdot (1+5\delta)|\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)| \\
&\qquad \leq (1+\beta)(1+5\delta) \cdot \frac{1}{\widehat{t}} \cdot \sum_{j \in \mathcal{L}} \sum_{v \in \mathrm{B}'_j} |\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v)| \,. \\
&\qquad \leq (1+2\beta) \cdot \frac{1}{\widehat{t}} \cdot \left(\sum_{j \in \mathcal{L}} \sum_{v \in \mathrm{B}'_j} |\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)| + \sum_{j \in \mathcal{L}} \sum_{v \in \mathrm{B}'_j} |\mathrm{Tr}_{\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}},[k]\setminus\widehat{\mathcal{L}}}(v) \setminus \mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)| \right) \\
&\qquad \leq (1+2\beta) \cdot \left(\frac{\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}}{\widehat{t}} + \frac{\mathrm{Tr}_I}{\widehat{t}} \right) \leq (1+2\beta) \left(\frac{\mathrm{Tr}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}}{(1-\beta)|\mathcal{T}(B'_{\mathcal{L}})|} + \frac{2\beta|\mathcal{T}(G)|}{\widehat{t}} \right) \\
&\qquad \leq (1+3\beta)\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 24\beta \,.
\end{aligned} \tag{64}$$

Where we used the fact that if the assumptions in this claim holds, then from the proof of Claim 3.3.18 it holds that:

$$(1-\beta)|\mathcal{T}(B'_{\mathcal{L}})| \leq \widehat{t} \leq (1+50\beta)|\mathcal{T}(B'_{\mathcal{L}})| \,.$$

Plugging Equations (62), (63) and (64) into Equation (60), we get that

$$\Pr[\chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}] \leq \alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta \,, \tag{65}$$

as required. ■

Proof of Lemma 3.4.7: By Claim 3.4.9 and Claim 3.4.8 it holds that

$$\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 70\beta \leq \Pr[\chi^{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v) = 1] \leq \alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta.$$

Recall that $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} = \frac{1}{r}\sum \chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v)$.
If $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \geq 71\beta$ then by applying the multiplicative Chernoff bound on $\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ we get:

$$\begin{aligned}
\Pr\left[\frac{1}{r}\sum \chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} > (1+\beta)\left(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta\right)\right] &< \exp\left(-\frac{\beta^2}{3} \cdot \left(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta\right) \cdot r\right) \\
&< \exp\left(-\frac{\beta^3}{3} \cdot \frac{20\log n}{\beta^3}\right) < \frac{1}{4n^3},
\end{aligned}$$

and similarly

$$\Pr\left[\frac{1}{r}\sum \chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} < (1-\beta)\left(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 70\beta\right)\right] < \exp\left(-\frac{\beta^2}{2}\cdot\left(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 70\beta\right)\cdot r\right)$$
$$< \exp\left(-\frac{\beta^3}{2}\cdot\frac{20\log n}{\beta^3}\right) < \frac{1}{4n^3}\,.$$

Therefore if $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \geq 71\beta$, then with probability at least $1-\frac{1}{2n^3}$

$$(1-\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 70\beta) \leq \widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leq (1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} + 70\beta). \tag{66}$$

If $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leq 70\beta$ then

$$\Pr\left[\frac{1}{r}\sum \chi^i_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}(v) > (1+\beta)\cdot 140\beta]\right] < \exp\left(-\frac{\beta^2}{3}\cdot 140\beta\cdot r\right) < \exp\left(-\frac{\beta^3}{3}\cdot\frac{20\log n}{\beta^3}\right) < \frac{1}{2n^3}.$$

Therefore, if $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leq 70\beta$, then with probability at least $1-1/2n^3$,

$$\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \leq 150\beta. \tag{67}$$

The first item of the claim follows from Equations (66) and (67). The second item of the claim follows in an almost identical manner. ■

Theorem 3.3 *Algorithm* Approx-Triangles-With-Advice$\langle G, \overline{\Delta}, \overline{m}, \epsilon\rangle$ *returns* $\widehat{\Delta}$ *such that, with probability at least* $1-\frac{1}{\log^3 n}$,

$$(1-\epsilon)\Delta(G) \leq \widehat{t} \leq (1+\epsilon)\Delta(G).$$

Proof: Recall the set of bad events defined in Claim 3.3.18:

1. E_1 – The selection of the safety intervals in Create-Random-Thresholds was not a good selection, where a good selection is as defined in Definition 3.3.1.
2. E_2 – Approx-Triangles-Degree(v,j) did not answer correctly on some sampled vertex, where "answering correctly" is as defined in Definition 3.3.6.
3. E_3 – Algorithm $\frac{1}{3}$-Approx-Triangles did not estimate the buckets' sizes correctly, as defined in Definition 3.3.13.

We define two additional bad events:

4. E_4 – Any of iterations of the loop in Step (2) of the algorithm Approx-Triangles-With-Advice was unsuccessful, as defined in Definition 3.4.4.
5. E_5 – $\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ is a not a good estimation of $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ or $\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ is not a good estimation of $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$, where good estimations are as defined in Definition 3.4.5.

As shown in the proof of Claim 3.3.18, with probability at least $1-\frac{1}{2\log^3 n}$ events E_1-E_3 does not occur. By Claim 3.4.3, if events E_1-E_3 do not occur, then event E_4 occurs with probability at most

$\frac{1}{n^2}$. By Claim 3.4.7, if events E_1-E_4 do not occur, then event E_5 occurs with probability at most $1/n^3$. Therefore, the following discussion holds with probability at least $1 - \frac{1}{\log^3 n}$.

Consider first the case that $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \geq 71\beta$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \geq 71\beta$. By Claims 3.3.18, 3.4.7 and 3.4.1 it holds that

$$\begin{aligned}
\widehat{\Delta} &= \frac{1}{3}\widehat{t}\cdot(1+2\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \leq \frac{1}{3}\widehat{t}\cdot\left(1+2(1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+70\beta)+\frac{1}{2}(1+\beta)(\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}+70\beta)\right)\\
&\leq \frac{1}{3}\widehat{t}\cdot(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}})+70\beta\widehat{t}\\
&\leq \frac{1}{3}(|\mathcal{T}(B'_{\mathcal{L}})|+50\beta|\mathcal{T}(G)|)\cdot(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}})+70\beta(1+50\beta)|\mathcal{T}(G)|\\
&\leq \frac{1}{3}(|\mathcal{T}(G)|+450\beta)|\mathcal{T}(G)|,
\end{aligned}$$

and that

$$\begin{aligned}
\widehat{\Delta} &\geq \frac{1}{3}\widehat{t}\cdot\left(1+2(1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}-70\beta)+\frac{1}{2}(1+\beta)(\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}-70\beta)\right)\\
&\geq \frac{1}{3}\widehat{t}\cdot(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}})-100\beta\widehat{t}\\
&\geq \frac{1}{3}(1-300\beta)|\mathcal{T}(G)|.
\end{aligned}$$

Putting the above together we get

$$\frac{1}{3}(1-300\beta)|\mathcal{T}(G)| \leq \widehat{t} \leq \frac{1}{3}(1+450\beta)|\mathcal{T}(G)|. \tag{68}$$

Now assume that $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} \geq 71\beta$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} < 71\beta$.

$$\begin{aligned}
\widehat{\Delta} &= \frac{1}{3}\widehat{t}\cdot(1+2\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \leq \frac{1}{3}\widehat{t}\cdot\left(1+2(1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+70\beta)+(1+\beta)\cdot 75\beta\right)\\
&\leq \frac{1}{3}\widehat{t}\cdot(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}})+100\beta\widehat{t}\\
&\leq \frac{1}{3}|\mathcal{T}(G)|+70\beta|\mathcal{T}(G)| \leq \frac{1}{3}(1+300\beta)|\mathcal{T}(G)|.
\end{aligned}$$

As for the lower bound, note that if $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} < 71\beta$ then $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} - 71\beta < 0$. Therefore

$$\begin{aligned}
\widehat{\Delta} &= \frac{1}{3}\widehat{t}\cdot(1+2\widehat{\alpha}_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\widehat{\alpha}_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}) \geq \frac{1}{3}\widehat{t}\cdot\left(1+2(1+\beta)(\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}-70\beta)+\frac{1}{2}(\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}-71\beta)\right)\\
&\geq \frac{1}{3}\widehat{t}\cdot(1+2\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}+\frac{1}{2}\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}})-100\beta\widehat{t}\\
&\geq \frac{1}{3}(1-300\beta)|\mathcal{T}(G)|.
\end{aligned}$$

The analysis for the remaining cases, where either $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}} < 71\beta$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}} \geq 71\beta$ or both $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ are smaller than 71β is almost identical.

Therefore for every value of $\alpha_{\mathcal{L},\widehat{\mathcal{S}},\widehat{\mathcal{S}}}$ and $\alpha_{\mathcal{L},\mathcal{L},\widehat{\mathcal{S}}}$ Equation (68) holds. Setting $\beta = \epsilon/450$ we get that with probability at least $1 - \frac{1}{\log^3 n}$,

$$(1-\epsilon)|\mathcal{T}(G)| \leq \widehat{\Delta} \leq (1+\epsilon)|\mathcal{T}(G)|,$$

and the proof is complete. ■

3.5 Removing the need for a priori knowledge on $\Delta(G)$ and m

We start by giving the high level idea. So far, we assumed that we have estimations $\overline{\Delta}$ and $\overline{m}$ such that $\frac{\Delta(G)}{c_\Delta} \leq \overline{\Delta} \leq \Delta(G)$ and $\overline{m} \geq \frac{m}{c_m}$. In order to remove this assumption, we describe an iterative process over the possible values of $\overline{\Delta}$ and $\overline{m}$, which runs until we get a good estimation of $\Delta(G)$. Namely, for $\overline{\Delta}$ we consider the values $1, 2, 4, \ldots, n^3$ and for $\overline{m}$ the values $1, 2, 4, \ldots, n^2$, so that we get $O(\log^2 n)$ possible pairs. We prove that, with a small modification to Approx-Trianlges-With-Advice, for a pair $\langle \overline{\Delta}, \overline{m} \rangle$ the query complexity of the algorithm is

$$\left(\frac{n}{\overline{T}^{1/3}} + \min \left\{ \overline{m}, \frac{\overline{m}^{3/2}}{\overline{T}} \right\} \right) \cdot \mathrm{poly}(\log n, 1/\epsilon), \tag{69}$$

and that the output $\widehat{\Delta}$ of the algorithm is such that $\widehat{\Delta} \leq (1+\epsilon)\Delta$. That is, it is either an underestimation of $\Delta(G)$ or a good estimation of it. We define $\overline{q} = \max \left\{ \frac{n}{\overline{\Delta}^{1/3}}, \min \left\{ \overline{m}, \frac{\overline{m}^{3/2}}{\overline{\Delta}} \right\} \right\}$ and iterate over the possible pairs of $\overline{\Delta}$ and $\overline{m}$ to get increasing values of $\overline{q}$. If during the iterative process the algorithm outputs $\widehat{\Delta}$ such that $\widehat{\Delta} \geq (1+\epsilon)\overline{\Delta}$, we halt the process and return $\widehat{\Delta}$. We prove that this is indeed a good estimation of $\Delta(G)$ with probability at least $1 - \frac{1}{\mathrm{polylog}\, n}$. We also prove that the process halts when running with good estimations of the values of $\Delta(G)$ and m, with probability at least $1 - \frac{1}{\mathrm{poly}\, n}$. This implies that the query complexity will not exceed

$$\left(\frac{n}{\Delta(G)^{1/3}} + \min \left\{ m, \frac{m^{3/2}}{\Delta(G)} \right\} \right) \cdot \mathrm{poly}(\log n, 1/\epsilon) \ ,$$

with probability at least $1 - \frac{1}{\mathrm{polylog}\, n}$.

3.5.1 Complexity analysis for fixed $\overline{\Delta}$ and $\overline{m}$

The assumption that $\overline{m} \geq \frac{m}{c_m}$ was only used directly in the proof of Claim 3.3.7 regarding the query complexity and running time of Approx-High-Triangles. To obtain a query complexity and running time as stated in Equation (69) we modify the procedure as follows.

Definition 3.5.1 *Let Approx-High-Triangles'' operate exactly as the procedure Approx-High-Triangles' operates except for the following modification. If Approx-High-Triangles' computes $\widehat{\Gamma}_{hi}$ that is greater than $(1+\delta) \cdot 2c_m \cdot \sqrt{\overline{m}}$, then the procedure aborts, causing Approx-Triangles-Degree and Approx-Triangles-With-Advice to abort as well.*

We first prove that with probability at least $1 - \frac{1}{\mathrm{poly}\, n}$ the procedure does not abort for values of $\overline{m}$ such that $\overline{m} \geq \frac{m}{c_m}$, and then proceed to prove that the above modification gives the desired running time.

Recall from Proposition 2.2 that for every graph G it holds that $\Delta(G) \leq m^{3/2}$. Therefore we can consider only pairs $\langle \overline{\Delta}, \overline{m} \rangle$ such that $\overline{m} \geq \overline{\Delta}^{2/3}$.

Claim 3.5.1 *For every $\overline{\Delta} \leq \overline{m}^{3/2}$, vertex v and index $j \in [k']$, if invoked with a value of $\overline{m}$ such that $\overline{m} \geq \frac{m}{c_m}$, then Approx-High-Triangles''$(v, j, d(v))$ aborts with probability at most $\frac{1}{2n^3}$.*

Proof: Recall from the proof of Claim 3.3.6, that if $|\Gamma_{hi}(v)| \geq \sqrt{\delta \cdot \mu_{j-1}}$, then with probability at least $1 - \frac{1}{4n^3}$,

$$(1-\delta) \cdot |\Gamma_{hi}(v)| \leq \widehat{\Gamma}_{hi} \leq (1+\delta) \cdot |\Gamma_{hi}(v)|.$$

Therefore if $|\Gamma_{hi}(v)| \geq \sqrt{\delta \cdot \mu_{j-1}}$, then with probability at least $1 - \frac{1}{4n^3}$, it holds that

$$\widehat{\Gamma}_{hi}(v) \leq (1+\delta) \cdot |\Gamma_{hi}(v)| \leq (1+\delta)\frac{2c_m \cdot m}{\sqrt{\overline{m}}} \leq 2(1+\delta)\sqrt{m}\,,$$

and the procedure does not abort.

If $|\Gamma_{hi}(v)| < \sqrt{\delta \cdot \mu_{j-1}}$, then by the choice of $s = \frac{d(v)}{\sqrt{\delta \cdot \mu_{j-1}}} \cdot \frac{20 \log n}{\delta^2}$ in Approx-High-Triangles and by the multiplicative Chernoff bound,

$$\Pr\left[\frac{1}{s}|S \cap \Gamma_{hi}| > (1+\delta) \cdot \frac{\sqrt{\delta \cdot \mu_{j-1}}}{d(v)}\right] < \exp\left(-\frac{\delta^2}{3} \cdot \frac{\sqrt{\delta \cdot \mu_{j-1}}}{d(v)} \cdot s\right) < \frac{1}{8n^3}\,.$$

Note that the procedure is invoked only with indices j such that $j \leq k' = \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1)}{\beta} \overline{T}^{2/3}$. Therefore, for every j, $\mu_j \leq (1+\beta)^{k'} \leq \frac{5c_\Delta(k+1)}{\beta} \cdot \overline{m}$, implying that $\sqrt{\delta \cdot \mu_{j-1}} \leq 2c_m \cdot \sqrt{\overline{m}}$. Hence, with probability at least $1 - \frac{1}{8n^3}$,

$$\widehat{\Gamma}_{hi} \leq (1+\delta) \cdot \sqrt{\delta \cdot \mu_{j-1}} \leq (1+\delta)2c_m \cdot \sqrt{\overline{m}}\,,$$

and the procedure does not abort. ■

Claim 3.5.2 *For every $\overline{m}$, the query complexity and running time of Approx-High-Triangles'', as defined in Definition 3.5.1, are $\frac{d(v) \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon)$.*

Proof: If $\overline{m} \geq \frac{m}{c_m}$ then the proof of Claim 3.3.7 holds and the claim follows.

If $\overline{m} < \frac{m}{c_m}$ then Equation (18) does not hold, and therefore Equation (19) does not follow. However, with the modification described in Definition 3.5.1, if the procedure reaches Step (8), then it did not abort, implying that $\widehat{\Gamma}_{hi} \leq (1+\delta)2c_m \cdot \sqrt{\overline{m}}$. Therefore, Equation (19) holds by the modification, and the claim follows as before. ■

Corollary 3.5.3 *For every v and j the query complexity and running time of procedures Approx-Triangles-Degree(v,j) and Sample-Random-Triangle(v,j) with the modified procedures Approx-High-Triangles'', Approx-Triangles-Of-Low-Deg-Vertices' and Approx-Crossing-Triangles' are $\frac{d(v) \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \mathrm{poly}(\log n, 1/\epsilon)$.*

To ensure that the query complexity is as stated in Equation (69) we also need to slightly modify the algorithms $\frac{1}{3}$-Approx-Triangles and Approx-Triangles-With-Advice. The main issue is that $\overline{m}$ might be smaller than the real value m, causing the running time to be higher than desired. Therefore we would like to modify the algorithms so that if they detects a "violation" to the values that they expect they abort.

The modified version of $\frac{1}{3}$-Approx-Triangles works as follows. After sampling the set of vertices S_j in Step (5), the modified algorithm queries the degree of all the sampled vertices. For $\ell = 0, \ldots, \log n$, denote by $S_{j,\ell}$ the subset of vertices in S_j with degree in $(2^{\ell-1}, 2^\ell]$. For every vertex

degree 2^ℓ there are at most $\frac{2m}{2^\ell}$ vertices of that degree. Therefore for every $j \in [k']$ and for every $\ell = 0, \ldots, \log n$ the expected number of vertices in $S_{j,\ell}$ is

$$\mathbb{E}[|S_{j,\ell}|] \leq s_j \cdot \frac{2m}{n \cdot 2^\ell} .$$

We modify $\frac{1}{3}$-Approx-Triangles so that if for some $S_{j,\ell}$ it detects a discrepancy between $|S_{j,\ell}|$ and its expected value then it aborts, causing Approx-Triangles-With-Advice to abort as well. Namely, $\frac{1}{3}$-Approx-Triangles aborts if $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} \geq \beta$ and $|S_{j,\ell}| \geq (1+\beta) s_j \cdot \frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell}$, or if $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} < \beta$ and $|S_{j,\ell}| \geq (1+\beta)\beta \cdot s_j$.

Similarly, let $S_{j_i,\ell}$ be the set of vertices sampled in Step (4) of Approx-Triangles-With-Advice, and denote by $S_{j_i,\ell}$ the vertices v with $d(v) \in (2^{\ell-1}, 2^\ell]$. As before,

$$\mathbb{E}[|S_{j_i,\ell}|] \leq s_{j_i} \cdot \frac{2m}{n \cdot 2^\ell} .$$

Therefore we modify Approx-Triangles-With-Advice to query the degree of every sampled vertex in Step (4) and keep track of the sizes of $S_{j_i,\ell}$ as more vertices are being sampled. Again, if at any step of the loop in Step (4) the algorithm detects a discrepancy as described before, it aborts.

Finally, in order to ensure that $\frac{1}{3}$-Approx-Triangles and Approx-Triangles-With-Advice will not perform more than $\left(\frac{n}{\overline{T}^{1/3}} + \min\left\{\overline{m}, \frac{\overline{m}^{3/2}}{\overline{T}}\right\}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$ queries we can do as follows. If the algorithms are invoked with values $\overline{\Delta}$ and $\overline{m}$ such that $\Delta < \sqrt{m}$, then instead of performing pair queries, the algorithms simply query for all the neighbors of the two vertices in the pair. Observe that we can always assume that the algorithms do not perform queries they can answer themselves. That is, we can allow the algorithms to save all the information they obtained from past queries, and assume they do not query for information they can deduce from their past queries. If in this setting the algorithms detect more than $2 \cdot \overline{m}$ edges, they abort.

We formalize the above discussion with the following definitions.

Definition 3.5.2 *Let $\frac{1}{3}$-Approx-Triangles' be a modified version of $\frac{1}{3}$-Approx-Triangles, which is defined as follows.*

- *After sampling the set of vertices S_j in Step (5), the modified algorithm queries the degree of all the sampled vertices, and for every $\ell = 0, \ldots, \log n$, defines the set $S_{j,\ell}$ to be the subset of vertices in S_j with degree in $(2^{\ell-1}, 2^\ell]$. If $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} \geq \beta$ and $|S_{j,\ell}| \geq (1+\beta) s_j \cdot \frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell}$, or if $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} < \beta$ and $|S_{j,\ell}| \geq (1+\beta)\beta \cdot s_j$ then the algorithm aborts.*
- *If the algorithm is invoked with values $\overline{\Delta}$ and $\overline{m}$ such that $\overline{\Delta} < \sqrt{\overline{m}}$ then instead of performing pair queries, the algorithm queries for all the neighbors of the two vertices in the pair. If the algorithm detects more than $2 \cdot \overline{m}$ edges, then it aborts.*

Definition 3.5.3 *Let Approx-Triangles-With-Advice' be modified version of Approx-Triangles-With-Advice, where the modifications are defined as follows.*

- *Approx-Triangles-With-Advice' invokes $\frac{1}{3}$-Approx-Triangles', instead of invoking $\frac{1}{3}$-Approx-Triangles.*
- *As more vertices are being sampled for the set S_{j_i} in Step (4), the modified algorithm queries the degree of all the sampled vertices, and for every $\ell = 0, \ldots, \log n$, defines the set $S_{j_i,\ell}$ to be the subset of vertices in S_{j_i} with degree in $(2^{\ell-1}, 2^\ell]$. If $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} \geq \beta$ and $|S_{j_i,\ell}| \geq (1+\beta) \cdot s_{j_i} \cdot \frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell}$, or if $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} < \beta$ and $|S_{j_i,\ell}| \geq (1+\beta)\beta \cdot s_{j_i}$ then the algorithm aborts.*

- *If the algorithm is invoked with values $\overline{\Delta}$ and $\overline{m}$ such that $\overline{\Delta} < \sqrt{\overline{m}}$ then instead of performing pair queries, the algorithm queries for all the neighbors of the two vertices in the pair. If the algorithm detects more than $2 \cdot \overline{m}$ edges, then it aborts.*

Claim 3.5.4 *For every $\overline{m} \geq \frac{m}{c_m}$ Approx-Triangles-With-Advice'$(G, \overline{\Delta}, \overline{m}, \epsilon)$ aborts with probability at most $\frac{1}{n^2}$.*

Proof: By Claim 3.5.1, for values $\overline{m} \geq \frac{m}{c_m}$, in each invocation, the procedure Approx-High-Triangles" aborts with probability at most $1 - \frac{1}{2n^3}$. Therefore during the entire run of Approx-Triangles-With-Advice' the procedure aborts with probability at most $\frac{1}{2n^2}$.

Now consider the invocation of $\frac{1}{3}$-Approx-Triangles'. For every j and ℓ, let $\chi_1^{j,\ell}, \ldots, \chi_{s_j}^{j,\ell}$ be Bernoulli random variables such that $\chi_i^{j,\ell} = 1$ if and only if the i^{th} sampled vertex in S_j has degree in the range $(2^{\ell-1}, 2^\ell]$. First consider ℓ such that $\frac{2m}{n \cdot 2^\ell} \geq \beta$. Since $\overline{m} \geq \frac{m}{c_m}$ it follows that $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} \geq \beta$ and the probability the algorithm will abort is

$$\begin{aligned}
\Pr\left[\frac{1}{s_j}\sum_{i=1}^{s_j}\chi_i^{j,\ell} \geq (1+\beta)\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell}\right] &\leq \Pr\left[\frac{1}{s_j}\sum_{i=1}^{s_j}\chi_i^{j,\ell} \geq (1+\beta)\frac{2m}{n \cdot 2^\ell}\right] \leq \exp\left(-\frac{\beta^2}{3} \cdot s_j \cdot \frac{2m}{n \cdot 2^\ell}\right) \\
&\leq \exp\left(-\frac{\beta^2}{3} \cdot \frac{2m}{n \cdot 2^\ell} \cdot \frac{n \cdot \mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n \cdot (k+1)}{\beta^3}\right) \\
&\leq \exp\left(-6 \log n \cdot (k+1)\right) \leq \frac{1}{2n^3}.
\end{aligned}$$

Now consider ℓ such that $\frac{2m}{n \cdot 2^\ell} < \beta$. If $\frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} \geq \beta$ then the algorithm aborts if $|S_{j,\ell}| > (1+\beta) \cdot s_j \cdot \frac{2c_m \cdot \overline{m}}{n \cdot 2^\ell} \geq (1+\beta)\beta \cdot s_j$. If $\frac{2\overline{m}}{n \cdot 2^\ell} < \beta$ then the algorithm aborts if $|S_{j,\ell}| > (1+\beta)\beta \cdot s_j$. Therefore the probability that the algorithm aborts is at most

$$\begin{aligned}
\Pr\left[\frac{1}{s_j}\sum_{i=1}^{s_j}\chi_i^{j,\ell} \geq (1+\beta)\beta\right] &\leq \exp\left(-\frac{\beta^2}{2} \cdot \beta \cdot \frac{n \cdot \mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n \cdot (k+1)}{\beta^3}\right) \\
&\leq \exp\left(-5 \log n \cdot (k+1)\right) \leq \frac{1}{2n^3}.
\end{aligned}$$

Since there are $(k'+1) \cdot (\log \overline{m} + 1) = O(\log n)$ indices $j \in [k']$ and $\ell \in [\log \overline{m}]$, it follows that $\frac{1}{3}$-Approx-Triangles aborts with probability at most $1 - \frac{1}{2n^2}$.

An almost identical analysis proves that the probability that Approx-Triangles-With-Advice' aborts during the loop in Step (4) is at most $\frac{1}{2n^2}$. Therefore, the probability that Approx-Triangles-With-Advice' aborts during its entire run is at most $\frac{1}{n^2}$. ■

Lemma 3.5.5 *For every $\overline{\Delta}$ and $\overline{m}$, the query complexity of Approx-Triangles-With-Advice'$(G, \overline{\Delta}, \overline{m}, \epsilon)$ is*

$$\left(\frac{n}{\overline{\mathcal{T}}^{1/3}} + \min\left\{\overline{m}, \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}}\right\}\right) \cdot \text{poly}(\log n, 1/\epsilon),$$

and the running time is

$$\left(\frac{n}{\overline{\mathcal{T}}^{1/3}} + \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}}\right) \cdot \text{poly}(\log n, 1/\epsilon).$$

Proof: It is clear that the running time of the algorithm is of the same order as the number of queries performed by it. Therefore, in the following discussion we only analyze the query complexity.

We first consider the case that $\overline{\Delta} \geq \sqrt{\overline{m}}$, and start by analyzing the query complexity of $\frac{1}{3}$-Approx-Triangles', invoked by Approx-Triangles-With-Advice' in Step (1). Fix an index $j \in [k']$. The query complexity of Steps (4) to (11) is as follows. In Step (5) the algorithm performs $s_j = \frac{n \cdot \mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n (k+1)}{\beta^3}$ queries. By Corollary 3.5.3 for every sampled vertex $v \in S_j$ the query complexity of Approx-Triangles-Deg(v, j) is $\frac{d(v)\sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon)$. For $\ell = 0, \ldots, \log n$, denote by $S_{j,\ell}$ the subset of vertices in S_j with degree in $(2^{\ell-1}, 2^\ell]$. By the definition of $\frac{1}{3}$-Approx-Triangles' (Definition 3.5.2), the query complexity for all the sampled vertices in S_j is upper bounded by

$$
\begin{aligned}
&\sum_{\ell}^{\log n} |S_{j,\ell}| \cdot \frac{2^\ell \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon) \\
&\quad = \sum_{\ell: \frac{2\overline{m}}{n \cdot 2^\ell} \geq \beta} |S_{j,\ell}| \cdot \frac{2^\ell \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon) + \sum_{\ell: \frac{2\overline{m}}{n \cdot 2^\ell} < \beta} |S_{j,\ell}| \cdot \frac{2^\ell \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon) \\
&\quad \leq \sum_{\ell: \frac{2\overline{m}}{n \cdot 2^\ell} \geq \beta} (1+\beta) s_j \cdot \frac{2\overline{m}}{n \cdot 2^\ell} \cdot \frac{2^\ell \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon) + \sum_{\ell: \frac{2\overline{m}}{n \cdot 2^\ell} < \beta} (1+\beta) s_j \cdot \beta \cdot \text{poly}(\log n, 1/\epsilon) \\
&\quad \leq \log n \cdot \frac{n \cdot \mu_j}{\overline{\mathcal{T}}} \cdot \frac{20 \log n \cdot (k+1)}{\beta^3} \cdot \frac{2\overline{m}}{n \cdot 2^\ell} \cdot \frac{2^\ell \cdot \sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon) \\
&\quad \leq \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}} \cdot \text{poly}(\log n, 1/\epsilon).
\end{aligned}
$$

Therefore, for every $j \in [k']$ the expected query complexity of Steps (4) to (11) is upper bounded by

$$\left(\frac{n \cdot \mu_j}{\overline{\mathcal{T}}} + \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}} \right) \cdot \text{poly}(\log n, 1/\epsilon) \, .$$

Since by definition $k' = \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1)}{\beta} \overline{\mathcal{T}}^{2/3}$, for every $j \in [k']$ it holds that $\mu_j \leq \frac{c_\Delta \cdot (k+1)}{\beta} \overline{\mathcal{T}}^{2/3}$. Hence, we have that for every $j \in [k']$ the expected query complexity of Steps (4) to (11) is

$$\left(\frac{n}{\overline{\mathcal{T}}^{1/3}} + \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}} \right) \cdot \text{poly}(\log n, 1/\epsilon).$$

Since there are at most $k' = O(\log \overline{\mathcal{T}}) = O(\log n)$ indices $j \in [k']$, the above is also the running time of Steps (3) to (11).

We now turn to analyze Steps (2) to (9) of Approx-Triangles-With-Advice'. By Corollary 3.5.3, for every sampled vertex v, invoking Sample-Random-Triangle(v, j_i) requires $\frac{d(v)\sqrt{\overline{m}}}{\mu_j} \cdot \text{poly}(\log n, 1/\epsilon)$ queries. Therefore, by a similar analysis to the above, for every j the number of queries performed in Step (4) of the algorithm is at most $\frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}} \cdot \text{poly}(\log n, 1/\epsilon)$.

For every sampled triangle $(v, u, w)_v$, invoking Is-Small-And-Significant(u) and Is-Small-And-Significant(w) requires $\frac{(d(v)+d(w)) \cdot \sqrt{\overline{m}}}{\overline{\mathcal{T}}^{2/3}} \cdot \text{poly}(\log n, 1/\epsilon)$ queries. Since the loop is repeated $\frac{20 \log n}{\beta^3} =$

poly$(1/\epsilon)$ times, and Steps (7) to and (9) take a constant time, the query complexity and running time of the entire algorithm are

$$\left(\frac{n}{\overline{\mathcal{T}}^{1/3}} + \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}}\right) \cdot \text{poly}(\log n, 1/\epsilon).$$

Now consider the case that $\overline{\Delta} < \sqrt{\overline{m}}$, so that $\overline{m} \leq \frac{\overline{m}^{3/2}}{\overline{\Delta}}$. In this setting, pair queries are replaced by querying for all the neighbors of the two vertices in the pair, and if at any point the algorithm detects more than $2\overline{m}$ edges then it aborts. Consider the vertices observed by the algorithm during its run. These vertices can be classified into two types. Vertices that are uniformly sampled by the algorithm among the set of all vertices, and vertices that are reached by a neighbor query. Observe that a vertex could be of both types. It follows from the above analysis that at most $\frac{n}{\overline{\mathcal{T}}^{1/3}}$ vertices are sampled directly, and it follows from the description of the modified version, that at most $2\overline{m}$ additional vertices are being observed through neighbor queries. Since the number of degree queries is upper bounded by the number of vertices observed by the algorithm, and the number of neighbor queries is upper bounded by $2 \cdot \overline{m}$, it follows that the query complexity in this case is $O\left(\frac{n}{\overline{\mathcal{T}}^{1/3}} + 2\overline{m}\right)$.

Therefore, for every pair of values $\overline{\Delta}$ and $\overline{m}$, the number of queries performed by the algorithm is

$$\left(\frac{n}{\overline{\mathcal{T}}^{1/3}} + \min\left\{\overline{m}, \frac{\overline{m}^{3/2}}{\overline{\mathcal{T}}}\right\}\right) \cdot \text{poly}(\log n, 1/\epsilon).$$

The running time is bounded as in the case that $\overline{\Delta} \geq \sqrt{\overline{m}}$. ■

Claim 3.5.6 *If Approx-Triangles-With-Advice' does not abort, then the following holds with probability* $1 - \frac{1}{\log^3 n}$.

1. *If* $\overline{\Delta}$ *is such that* $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \Delta$ *and the algorithm does not abort, then the output* $\widehat{\Delta}$ *of the algorithm is such that* $(1-\epsilon)\Delta(G) \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G)$.
2. *For every* $\overline{\Delta}$ *and* $\overline{m}$, *the output of the algorithm* $\widehat{\Delta}$ *is such that* $\widehat{\Delta} \leq (1+\epsilon)\Delta(G)$.

Proof: Observe that in the previous subsections, the assumption that $\overline{m} \geq \frac{m}{c_m}$ was only used in the proof of the claim regarding the running time of the procedure Approx-High-Triangles (Claim 3.3.7), and not in any of the proofs regarding the correctness of the output of the algorithm. Therefore the only affect on the correctness of the output could stem from the use of the modified procedure Approx-High-Triangles" described in Definition 3.5.1. By Definition 3.5.1, the procedure operates exactly as Approx-High-Triangles', except that if for some vertex v it computes $\widehat{\Gamma}_{hi}$ such that $\widehat{\Gamma}_{hi} \geq 2(1+\delta)\sqrt{\overline{m}}$ then it aborts, and consequently Approx-Triangles-With-Advice' aborts. This implies that if Approx-Triangles-With-Advice' does not abort, then the removal of the assumption on $\overline{m}$ does not affect the correctness of the estimation of $|\mathcal{T}(G)|$. Therefore, if $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \Delta$, then for any value of $\overline{m}$, if the algorithm does not abort, then it outputs $\widehat{\Delta}$ such that $(1-\epsilon)\Delta(G)\widehat{\Delta} \leq (1+\epsilon)\Delta(G)$.

The assumption that $\overline{\Delta} \leq \Delta(G)$ is directly used only in Claims 3.2.1 and 3.3.12 and the assumption that $\overline{\Delta} \geq \Delta(G)/c_\Delta$ is directly used only in Claim 3.3.13. Therefore, if we remove the assumption on $\overline{\Delta}$, then the above claims no longer hold. If $\overline{\Delta} > \Delta$ then the number of triangles in

$Tr_{\mathcal{S}^*,\mathcal{S}^*,\mathcal{S}^*}$ and in Tr_I is no longer bounded by an order of $\beta|\mathcal{T}(G)|$. Hence, ignoring these buckets might lead to an underestimation of $|\mathcal{T}(G)|$, but it may not cause an overestimation.

If $\overline{\Delta} < \frac{\Delta}{c_\Delta}$ then there may be large significant buckets for which $\mu_j > \frac{c_\Delta \cdot (k+1) \cdot \overline{T}^{2/3}}{\beta}$. Since we only let j run up to $k' = \log_{(1+\beta)} \frac{c_\Delta \cdot (k+1) \cdot \overline{T}^{2/3}}{\beta}$ the algorithm does not estimate the sizes of these buckets. This again could lead to an underestimation of $|\mathcal{T}(G)|$, but not to an overestimation. Hence $\widehat{\Delta} \leq (1+\epsilon)\Delta$.

For values of $\overline{\Delta}$ and $\overline{m}$ such that $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \Delta$ and $\overline{m} \geq \frac{m}{c_m}$, Assumption 2.1 holds. Therefore, if the algorithm does not abort, then it returns $\widehat{\Delta}$ such that $(1-\epsilon)\Delta \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G)$ with probability at least $1 - \frac{1}{\log^3 n}$. ■

3.5.2 The search for $\overline{\Delta}$ and $\overline{m}$: The Approx-Triangles algorithm

Algorithm 10 Approx-Triangles(G, ϵ)

```
Let q̂ ← 1.
While q̂ ≤ n^2 do
    For q̄ = 1, 2, 4 ..., q̂ do
        For i = 0, 1, 2, 3, ..., log(q̄^2) do
            Let Δ̄_{q̄,i} = n^3/(q̄·2^i), m̄_{q̄,i} = max{ q̄, n^2/2^{2i/3} } and let p_{q̄,i} = ⟨Δ̄_{q̄,i}, m̄_{q̄,i}⟩.
            Let Δ̂_{q̄,i} ← Approx-Triangles-With-Advice'(G, Δ̄_{q̄,i}, m̄_{q̄,i}, ε).
            If Approx-Triangles-With-Advice'(G, Δ̄_{q̄,i}, m̄_{q̄,i}, ε) aborted then
                q̂ ← 2q̂.
                Go to Step (3).        ▷ In this case we say that the algorithm skips.
            If Δ̂ ≥ (1+ε)Δ̄ then
                Return Δ̂.
            Continue to the next pair.  ▷ In this case we say that the algorithm continues.
        End for
    End for
    Let q̂ ← 2q̂.
End while
```

Theorem 3.4 *Algorithm Approx-Triangles returns $\widehat{\Delta}$ such that with probability at least 2/3,*

$$(1-\epsilon)\Delta(G) \leq \widehat{\Delta} \leq (1+\epsilon)\Delta(G).$$

The expected query complexity of Approx-Triangles is

$$\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{ m, \frac{m^{3/2}}{\Delta(G)} \right\} \right) \cdot \mathrm{poly}(\log n, 1/\epsilon) \ ,$$

and the expected running time is

$$\left(\frac{n}{\Delta(G)^{1/3}} + \frac{m^{3/2}}{\Delta(G)} \right) \cdot \mathrm{poly}(\log n, 1/\epsilon) \ .$$

Moreover, for every k, the probability that the running and query complexity exceeds 2^k times their expected values is at most $(\frac{1}{\log^2 n})^k$.

In what follows we prove Theorem 3.4, but we start by briefly explaining one aspect of the algorithm which may seem redundant. Namely, a natural question is why does the algorithm have both an outer loop over $\widehat{q}$ and an inner loop over $\overline{q} \leq \widehat{q}$, rather than just one loop over increasing values of $\overline{q}$. Indeed we can prove that once the algorithm considers a value $\overline{q}$ that is within a constant factor of $q = \max\left\{\frac{n}{\Delta(G)^{1/3}}, \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right\}$, then with high probability it will output a good estimate of $\Delta(G)$, as desired. We can also prove that as long as $\overline{q}$ is significantly smaller than q, then the probability that the algorithm outputs a bad estimate of $\Delta(G)$ is sufficiently small. Furthermore, the accumulated query complexity and running time of the algorithm (over all $\overline{q} = O(q)$) is sufficiently small. However, once $\overline{q}$ "passes" q, which may occur with a small probability, we do not know how to bound the probability that the algorithm terminates. Therefore, we have an outer loop over increasing values of $\widehat{q}$, but for each such value, we "reconsider" all values $\overline{q} \leq \widehat{q}$. This ensures that if we reach $\widehat{q} > q$, we still call Approx-Triangles-With-Advice' with a "correct" setting of $\langle\overline{\Delta}, \overline{m}\rangle$, which corresponds to $\overline{q} = \Theta(q)$. Hence, the probability that the algorithm does not terminate after $\widehat{q}$ becomes larger than q (if this occurs), decreases exponentially with the number of different settings of $\widehat{q}$.

We start by proving the following claim regarding the relationship between $\overline{q}$ and the pairs $\langle\overline{\Delta}_{\overline{q}}, \overline{m}_{\overline{q}}\rangle$. For the sake of brevity, in what follows let $\Delta = \Delta(G)$.

Claim 3.5.7 *For every $\overline{q}$ and every $0 \leq i \leq \log(\overline{q}^2)$, let $\overline{\Delta}_{\overline{q},i}$ and $\overline{m}_{\overline{q},i}$ be as defined in Step (5) of Approx-Triangles. For every $0 \leq i \leq \log(\overline{q}^2)$,*

$$\max\left\{\frac{n}{\overline{\Delta}_{\overline{q},i}^{1/3}}, \min\left\{\overline{m}_{\overline{q},i}, \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}}\right\}\right\} = \overline{q}\ .$$

Proof: By the setting of $\overline{\Delta}_{\overline{q},i}$ in Approx-Triangles, for every $0 \leq i \leq \log(\overline{q}^2)$, $\overline{\Delta}_{\overline{q},i} = \frac{n^3}{\overline{q}\cdot 2^i}$. Therefore,

$$\frac{n}{\overline{\Delta}_{\overline{q},i}^{1/3}} = (\overline{q}\cdot 2^i)^{1/3} \leq (\overline{q}\cdot\overline{q}^2)^{1/3} \leq \overline{q}\ . \tag{70}$$

Therefore, in order to prove the claim it is sufficient to prove that for every $i \in [\log(\overline{q}^2)]$, it holds that $\min\left\{\overline{m}_{\overline{q},i}, \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}}\right\} = \overline{q}$.

By the setting of $\overline{m}_{\overline{q},i}$ in Approx-Triangles, for every $i \in [\log(\overline{q}^2)]$, $\overline{m}_{\overline{q},i} = \max\left\{\overline{q}, \frac{n^2}{2^{2i/3}}\right\}$, and there are two possibilities. First consider the case that $\overline{q} \geq \frac{n^2}{2^{2i/3}}$, so that $\overline{m}_{\overline{q},i} = \overline{q}$. In this case,

$$2^i \geq \frac{n^3}{\overline{q}^{3/2}}\ , \tag{71}$$

Implying that

$$\frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}} = \frac{\overline{q}^{3/2}}{\overline{\Delta}_{\overline{q},i}} = \frac{\overline{q}^{3/2}}{\frac{n^3}{\overline{q}\cdot 2^i}} = \frac{\overline{q}^{5/2}}{n^3}\cdot 2^i \geq \frac{\overline{q}^{5/2}}{n^3}\cdot\frac{n^3}{\overline{q}^{3/2}} = \overline{q}\ . \tag{72}$$

Therefore,

$$\min\left\{\overline{m}_{\overline{q},i}, \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}}\right\} = \overline{q}, \tag{73}$$

and together with Equation (70), the claim follows.

If $\overline{q} \leq \frac{n^2}{2^{2i/3}}$ then $\overline{m}_{\overline{q},i} = \frac{n^2}{2^{2i/3}}$. In this case, the direction of the inequality in Equation (72) is "flipped", and we get

$$\frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}} \leq \overline{q} \leq \frac{n^2}{2^{2i/3}} = \overline{m}_{\overline{q},i}\ . \tag{74}$$

The above, together with the setting of $\overline{\Delta}_{\overline{q},i}$ implies

$$\min\left\{\overline{m}_{\overline{q},i}, \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}}\right\} = \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}} = \frac{\left(\frac{n^2}{2^{2i/3}}\right)^{3/2}}{\frac{n^3}{\overline{q}\cdot 2^i}} = \overline{q}\ . \tag{75}$$

This completes the proof. ■

The following is a corollary of Lemma 3.5.5 and Claim 3.5.7.

Corollary 3.5.8 *For every $\overline{q}$ and every $0 \leq i \leq \log(\overline{q}^2)$, let $\overline{\Delta}_{\overline{q},i}$ and $\overline{m}_{\overline{q},i}$ be as defined in Step (5) of Approx-Triangles. The query complexity of* Approx-Triangles-With-Advice'*$(G, \overline{\Delta}_{\overline{q},i}, \overline{m}_{\overline{q},i}, \epsilon)$ is $\overline{q} \cdot \mathrm{poly}(\log n, 1/\epsilon)$, and the running time is $\max\left(\frac{n}{\overline{\Delta}_{\overline{q},i}^{1/3}} + \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}}\right) \cdot \mathrm{poly}(\log n, 1/\epsilon)$.*

Definition 3.5.4 *Let $\widehat{\Delta}$ be the returned value from the invocation of* Approx-Triangles-With-Advice' *in Step (6) of Approx-Triangles. If a value $\widehat{\Delta}$ is returned, and is such that $(1-\epsilon)\Delta \leq \widehat{\Delta} \leq (1+\epsilon)\Delta$, then we say that $\widehat{\Delta}$ is a* good estimate*, and otherwise we say that $\widehat{\Delta}$ is a* bad estimate*.*

Claim 3.5.9 *For every pair $\langle \overline{\Delta}_{\overline{q},i}, \overline{m}_{\overline{q},i} \rangle$ the following holds:*

1. *If $\overline{\Delta} > \Delta$ then Approx-Triangles either skips or continues with probability at least $1 - \frac{1}{\log^3 n}$, where "skips" and "continues" are as defined in Steps (9) and (12), respectively.*
2. *If $\frac{\Delta}{2} < \overline{\Delta} \leq \Delta$ then Approx-Triangles returns a bad estimate with probability at most $\frac{1}{\log^3 n}$.*
3. *If $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \frac{\Delta}{2}$ then there are two possibilities.*
   (a) *If $\overline{m} \geq \frac{m}{c_m}$, then with probability at least $1 - \frac{2}{\log^3 n}$, Approx-Triangles returns a good estimate.*
   (b) *Otherwise, with probability at least $1 - \frac{1}{\log^3 n}$, Approx-Triangles either skips or returns $\widehat{\Delta}$ which is a good estimate.*

Proof: Let $\widehat{\Delta}$ be the returned value of Approx-Triangles-With-Advice'$(G, \overline{\Delta}, \overline{m}, \epsilon)$.

1. Let $\overline{\Delta}$ be such that $\overline{\Delta} > \Delta$. By Item (2) in Claim 3.5.6, for every values of $\overline{\Delta}$ and $\overline{m}$, it holds that $\widehat{\Delta} \leq (1+\epsilon)\Delta$ with probability at least $1 - \frac{1}{\log^3 n}$. Therefore, the probability that $\widehat{\Delta} \geq (1+\epsilon)\overline{\Delta} > (1+\epsilon)\Delta$ is at most $\frac{1}{\log^3 n}$.

2. If $\frac{\Delta}{2} < \overline{\Delta} \leq \Delta$ then by Item (1) in Claim 3.5.6, if Approx-Triangles-With-Advice' does not abort, then $\widehat{\Delta}$ is a good estimate with probability at least $1 - \frac{1}{\log^3 n}$. Therefore Approx-Triangles returns a bad estimate with probability at most $\frac{1}{\log^3 n}$.

3. (a) Let $\langle \overline{\Delta}, \overline{m} \rangle$ be a pair for which $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \frac{\Delta}{2}$ and $\overline{m} \geq \frac{m}{c_m}$. Since $\overline{m} \geq \frac{m}{c_m}$, by Claim 3.5.4, Approx-Triangles-With-Advice' aborts with probability at most $\frac{1}{2n^3}$. Since $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \frac{\Delta}{2}$, by Item (1) in Claim 3.5.6, Approx-Triangles-With-Advice' returns a good estimate with probability at least $1 - \frac{2}{\log^3 n}$. Therefore, with probability at least $1 - \frac{2}{\log^3 n}$,

$$\widehat{\Delta} \geq (1-\epsilon)\Delta \geq 2(1-\epsilon)\overline{\Delta} \geq (1+\epsilon)\overline{\Delta},$$

and Approx-Triangles returns $\widehat{\Delta}$ which is a good estimate.

(b) If $\frac{\Delta}{c_\Delta} \leq \overline{\Delta} \leq \frac{\Delta}{2}$ and $\overline{m} < \frac{m}{c_m}$, then by Item (2) in Claim 3.5.6, Approx-Triangles-With-Advice' either aborts or it returns $\widehat{\Delta}$ which is a good estimate with probability at least $1 - \frac{1}{\log^3 n}$. If $\widehat{\Delta}$ is a good estimate then as before,

$$\widehat{\Delta} \geq (1-\epsilon)\Delta \geq (1-\epsilon) \cdot 2\overline{\Delta} \geq (1+\epsilon)\overline{\Delta},$$

and therefore Approx-Triangles returns $\widehat{\Delta}$. Hence we get that if Approx-Triangles-With-Advice' does not abort then Approx-Triangles returns a good estimate with probability at least $1 - \frac{1}{\log^3 n}$.

■

In order to complete the discussion we set $c_\Delta = 4$ and $c_m = 4$ and prove the following claim.

Claim 3.5.10 *Let* $q = \max\left\{\frac{n}{\Delta^{1/3}}, \min\left\{m, \frac{m^{3/2}}{\Delta}\right\}\right\}$. *For* $\widehat{q}$ *such that* $\widehat{q} < 2q$, *it holds that* $\frac{n^3}{\widehat{q}} > \frac{\Delta}{c_\Delta}$.

Proof: Since $q = \max\left\{\frac{n}{\Delta^{1/3}}, \min\left\{m, \frac{m^{3/2}}{\Delta}\right\}\right\}$, then either $q = \frac{n}{\Delta^{1/3}}$ or $q = \min\left\{m, \frac{m^{3/2}}{\Delta}\right\}$. In the first case $\Delta = \frac{n^3}{q^3}$ and

$$\frac{n^3}{\widehat{q}} > \frac{n^3}{2q} \geq \frac{n^3}{2q^3} \geq \frac{\Delta}{2} > \frac{\Delta}{c_\Delta}.$$

In the second case, $q = \min\left\{m, \frac{m^{3/2}}{\Delta}\right\}$ implying that $q \leq \frac{m^{3/2}}{\Delta}$. Hence,

$$\frac{n^3}{\widehat{q}} > \frac{n^3}{2q} \geq \frac{n^3}{2m^{3/2}/\Delta} \geq \frac{\Delta}{2} > \frac{\Delta}{c_\Delta}.$$

Therefore, in either case the claim holds. ■

Proof of Theorem 3.4: Let $q = \max\left\{\frac{n}{\Delta(G)^{1/3}}, \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right\}$, for every $\widehat{q} = 1, 2, 4, \ldots, n^2$ let $Q' = \{1, 2, 4 \ldots, \widehat{q}\}$, and for every $\overline{q} \in Q'$ let $p_{\overline{q},i}$ be as defined in Step (5) of Approx-Triangles so that $p_{\overline{q},i} = \left\langle \frac{n^3}{\overline{q} \cdot 2^i}, \max\left\{\frac{n^2}{n^{2i/3}}, \overline{q}\right\}\right\rangle$. Let $P_{\overline{q}} = \{p_{\overline{q},0}, \ldots, p_{\overline{q},\log(\overline{q}^2)}\}$.

1. Consider $\widehat{q}$ such that $\widehat{q} < 2q$. For every $\overline{q} \in Q'$, $\overline{q} \leq \widehat{q} < 2q$, and by Claim 3.5.10, $\frac{n^3}{\overline{q}} > \frac{\Delta}{c_\Delta}$. By the order of the pairs in $P_{\overline{q}}$, for each pair $\langle \overline{\Delta}_{\overline{q},i}, \overline{m}_{\overline{q},i} \rangle$ in $P_{\overline{q}}$, it holds that $\overline{\Delta}_{\overline{q},i} = \frac{n^3}{\overline{q} \cdot 2^i} \leq \frac{n^3}{\overline{q}}$, and there are two possibilities:

   (a) For all the pairs in $P_{\overline{q}}$, $\overline{\Delta}_{\overline{q},i} \geq \frac{\Delta}{c_\Delta}$. In this case, for every pair the probability that Approx-Triangles-With-Advice' returns a bad estimate is at most $\frac{1}{\log^3 n}$. Since for every value of $\overline{q}$ there are at most $2 \log n$ pairs in $P_{\overline{q}}$, the probability that Approx-Triangles returns a bad estimate for $\widehat{q} \leq 2q$ is at most $\frac{2 \log n}{\log^3 n}$.

   (b) The second possibility is that there exist pairs in $P_{\overline{q}}$ such that $\overline{\Delta}_{\overline{q},i} < \frac{\Delta}{c_\Delta}$. For these pairs we do not have a guarantee that Approx-Triangles returns a good estimate, but we show that with high probability we will not reach them. Observe that the order of the pairs in $P_{\overline{q}}$ implies a monotonic decreasing order on $\overline{\Delta}_{\overline{q},i}$. Let $p_{\overline{q},i^*}$ be the first pair in $P_{\overline{q}}$ for which $\overline{\Delta}_{\overline{q},i} < \frac{\Delta}{c_\Delta}$. By the order of the pairs in $P_{\overline{q}}$, if we set $c_\Delta = 4$, then it holds that there exists a pair $p_{\overline{q},j}$ for some $j < i^*$ for which $\frac{\Delta}{4} \leq \overline{\Delta}_{\overline{q},j} \leq \frac{\Delta}{2}$. By Item (3) of Claim 3.5.9, for the pair $p_{\overline{q},j}$, with probability at least $1 - \frac{1}{\log^3 n}$, Approx-Triangles either returns a good estimate or it skips, implying that all the following pairs in $P_{\overline{q}}$ are skipped. Therefore the probability of returning a bad estimate is at most $\frac{1}{\log^3 n}$.

   Therefore for every $\widehat{q} < 2q$, Approx-Triangles returns a bad estimate with probability at most $\frac{2 \log n}{\log^3 n} = \frac{1}{\log^2 n}$.

2. Now let $2q \leq \widehat{q} \leq 4q$. For $\overline{q} = \widehat{q}$ it holds that $2q \leq \overline{q} \leq 4q$, and since $\overline{\Delta}_{\overline{q},\log(\overline{q}^2)} \leq \frac{\Delta}{8}$, then there exists a pair $p^* = p_{\overline{q},i^*} = \langle \overline{\Delta}_{\overline{q},i^*}, \overline{m}_{\overline{q},i^*} \rangle$ in $P_{\overline{q}}$ such that $\frac{\Delta}{4} \leq \overline{\Delta}_{\overline{q},i^*} \leq \frac{\Delta}{2}$. By Equations (73) and (75) in Claim 3.5.7, $\overline{q} = \min\left\{\overline{m}_{\overline{q},i}, \frac{\overline{m}_{\overline{q},i}^{3/2}}{\overline{\Delta}_{\overline{q},i}}\right\}$ for every pair $\langle \overline{\Delta}_{\overline{q},i}, \overline{m}_{\overline{q},i} \rangle$ in $P_{\overline{q}}$. If $q = \frac{m^{3/2}}{\Delta}$ then

$$\frac{\overline{m}_{\overline{q},i^*}^{3/2}}{\overline{\Delta}_{\overline{q},i^*}} \geq \min\left\{\overline{m}_{\overline{q},i^*}, \frac{\overline{m}_{\overline{q},i^*}^{3/2}}{\overline{\Delta}_{\overline{q},i^*}}\right\} = \overline{q} \geq 2q = 2 \cdot \frac{m^{3/2}}{\Delta},$$

   and if $q = m$ then

$$\overline{m}_{\overline{q},i^*} \geq \min\left\{\overline{m}_{\overline{q},i^*}, \frac{\overline{m}_{\overline{q},i^*}^{3/2}}{\overline{\Delta}_{\overline{q},i^*}}\right\} = \overline{q} \geq 2q = 2m.$$

   It follows that for the pair $p_{\overline{q},i^*}$, $\overline{m}_{\overline{q},i^*} \geq \frac{m}{c_m}$. Hence, by Item (3a) in Claim 3.5.9, Approx-Triangle returns $\widehat{\Delta}$, which is a good estimate of Δ with probability at least $1 - \frac{2}{\log^3 n}$.

3. Finally, consider $\widehat{q}$ such that $\widehat{q} > 4q$. In this case we cannot bound the probability that Approx-Triangles returns a bad estimate. However, since for every $\widehat{q}$ Approx-Triangles first runs over all $\overline{q}$'s such that $\overline{q} \leq \widehat{q}$, it holds by the previous two items that for every $\widehat{q} > 4q$, the probability for returning a good estimate is at least $1 - \frac{2}{\log^2 n}$.

It follows from Items (1) to (3) that Approx-Triangles returns a good estimate with probability at least $1 - \frac{2 \log n}{\log^2 n} \geq \frac{2}{3}$. Hence, the first part of the theorem holds.

By Corollary 3.5.8, for every $\overline{q} \leq q$, the expected query complexity of Approx-Triangles-With-Advice' is $\overline{q} \cdot \mathrm{poly}(\log n, 1/\epsilon)$ Therefore for every $\widehat{q} \leq 4q$ the query complexity is bounded by

$O(q) \cdot \text{poly}(\log n, 1/\epsilon)$. For every $\widehat{q} > 4q$ the probability that Approx-Triangles will reach $\overline{q}$ such that $\overline{q} > 4q$ is at most $\frac{2}{\log^2 n}$. Therefore, the expected query complexity of Approx-Triangles is

$$\left(\frac{n}{\Delta(G)^{1/3}} + \min \left\{ m, \frac{m^{3/2}}{\Delta(G)} \right\} \right) \cdot \text{poly}(\log n, 1/\epsilon).$$

Similarly, the expected running time is

$$\left(\frac{n}{\Delta(G)^{1/3}} + \frac{m^{3/2}}{\Delta(G)} \right) \cdot \text{poly}(\log n, 1/\epsilon).$$

The probability that the query complexity and running time exceed 2^k times their expected values is at most $(\frac{1}{\log^2 n})^k$. ■

4 A Lower Bound

In this section we present a lower bound on the number of queries necessary for estimating the number of triangles in a graph. This lower bound matches our upper bound in terms of the dependence on n, m and $\Delta(G)$, up to polylogarithmic factors in n and the dependence in $1/\epsilon$. In what follows, when we refer to approximation algorithms for the number of triangles in a graph, we mean multiplicative-approximation algorithms that output with high constant probability an estimation $\widehat{\Delta}$ such that $\Delta(G)/C \leq \widehat{\Delta} \leq C \cdot \Delta(G)$ for some predetermined approximation factor C.

We consider multiplicative-approximation algorithms that are allowed the following three types of queries: Degree queries, pair queries and random new-neighbor queries. Degree queries and pair queries are as defined in Section 2. A random new-neighbor query q_i is a single vertex u and the corresponding answer is a vertex v such that $(u, v) \in E$ and the edge (u, v) is selected uniformly at random among the edges incident to u that have not yet been observed by the algorithm. In Corollary 4.1 we show that this implies a lower bound when the algorithm may perform (standard) neighbor queries instead of random new-neighbor queries.

We first give a simple lower bound that depends on n and $\Delta(G)$.

Theorem 4.1 *Any multiplicative-approximation algorithm for the number of triangles in a graph must perform* $\Omega\left(\frac{n}{\Delta(G)^{1/3}}\right)$ *queries, where the allowed queries are degree queries, pair queries and random new-neighbor queries.*

Proof: For every n and every $1 \leq \Delta \leq \binom{n}{3}$ we next define a graph G_1 and a family of graphs $\mathcal{G}_2$ for which the following holds. The graph G_1 is the empty graph over n vertices. In $\mathcal{G}_2$, each graph consists of a clique of size $\lfloor \Delta^{1/3} \rfloor$ and an independent set of size $n - \lfloor \Delta^{1/3} \rfloor$. See Figure 4 for an illustration. Within $\mathcal{G}_2$ the graphs differ only in the labeling of the vertices. By construction, G_1 contains no triangles and each graph in $\mathcal{G}_2$ contains $\Theta(\Delta)$ triangles. Clearly, unless the algorithm "hits" a vertex in the clique it cannot distinguish between the two cases. The probability of hitting such a vertex in a graph selected uniformly at random from $\mathcal{G}_2$ is $\lfloor \Delta^{1/3} \rfloor / n$. Thus, in order for this event to occur with high constant probability, $\Omega\left(\frac{n}{\Delta^{1/3}}\right)$ queries are necessary. ■

We next state our main theorem.

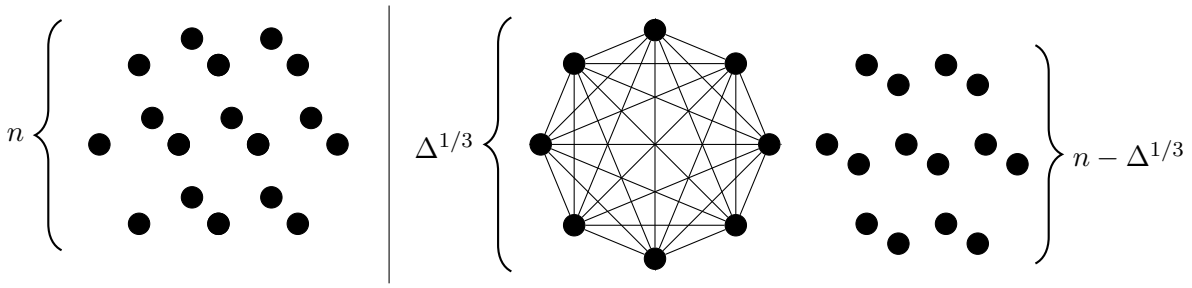


Figure 4: An illustration of the two families.

Theorem 4.2 *Any multiplicative-approximation algorithm for the number of triangles in a graph must perform* $\Omega\left(\min\left\{\frac{m^{3/2}}{\Delta(G)}, m\right\}\right)$ *queries, where the allowed queries are degree queries, pair queries and random new-neighbor queries.*

For every n, every $1 \leq m \leq \binom{n}{2}$ and every $1 \leq \Delta \leq \min\{\binom{n}{3}, m^{3/2}\}$ we define a graph G_1 and a family of graphs $\mathcal{G}_2$ for which the following holds. The graph G_1 and all the graphs in $\mathcal{G}_2$ have n vertices and m edges. For the graph G_1, $\Delta(G_1) = 0$, and for every graph $G \in \mathcal{G}_2$, $\Delta(G) = \Theta(\Delta)$. We prove it is necessary to perform $\Omega\left(\min\left\{\frac{m^{3/2}}{\Delta}, m\right\}\right)$ queries in order to distinguish with high constant probability between G_1 and a random graph in $\mathcal{G}_2$. For the sake of simplicity, in everything that follows we assume that $\sqrt{m}$ is even.

We prove that for values of Δ such that $\Delta < \frac{1}{4}\sqrt{m}$, at least $\Omega(m)$ queries are required, and for values of Δ such that $\Delta \geq \sqrt{m}$ at least $\Omega\left(\frac{m^{3/2}}{\Delta}\right)$ queries are required. We delay the discussion on the former case to Subsection 4.4, and start with the case that $\Delta \geq \sqrt{m}$. Our construction of $\mathcal{G}_2$ depends on the value of Δ as a function of m. We show three different constructions for the following ranges of Δ:

1. $\Delta = m$.
2. $m < \Delta \leq \frac{m^{3/2}}{8}$.
3. $\sqrt{m} \leq \Delta \leq \frac{m}{4}$.

We prove that for every Δ as above, $\Omega(\frac{m^{3/2}}{\Delta})$ queries are needed in order to distinguish between the graph G_1 and a random graph in $\mathcal{G}_2$.

We start by addressing the case that $\Delta = m$ in Subsection 4.1, and deal with the case that $m < \Delta \leq \frac{m^{3/2}}{8}$ in Subsection 4.2, and with the case that $\sqrt{m} \leq \Delta \leq \frac{m}{4}$ in Subsection 4.3.

Observe that by Proposition 2.2, for every graph G, it holds that $\Delta(G) \leq m^{3/2}$. Hence, the above ranges indeed cover all the possible values of Δ as a function of m.

Before embarking on the proof for $\Delta = m$, we introduce the notion of a *knowledge graph* (as defined previously in e.g., [GR02]), which will be used in all lower bound proofs. Let ALG be a triangles approximation algorithm that performs Q queries, let q_t denote its t^{th} query and let a_t denote the corresponding answer. Then ALG is a (possibly probabilistic) mapping from *query-answer histories* $\pi \stackrel{\text{def}}{=} \langle (q_1, a_1), \ldots, (q_t, a_t) \rangle$ to q_{t+1}, for every $t < Q$, and to $\mathbb{N}$ for $t = Q$.

We assume that the mapping determined by the algorithm is determined only on histories that are consistent with the graph G_1 or one of the graphs in $\mathcal{G}_2$. Any query-answer history π of length t can be used to define a knowledge graph G_π^{kn} at time t. Namely, the vertex set of G_π^{kn} consists of n vertices. For every new-neighbor query u_i answered by v_i for $i \leq t$, the knowledge graph contains the edge (u_i, v_i), and similarly for every pair query (u_j, v_j) that was answered by 1. In addition, for every pair query (u_i, v_i) that is answered by 0, the knowledge graph maintains the information that (u_i, v_i) is a non-edge. The above definition of the knowledge graph is a slight abuse of the notation of a graph since G_π^{kn} is a subgraph of the graph tested by the algorithm, but it also contains additional information regarding queried pairs that are not edges. For a vertex u, we denote its set of neighbors in the knowledge graph by $\Gamma_\pi^{kn}(u)$, and let $d_\pi^{kn}(u) = \left|\Gamma_\pi^{kn}(u)\right|$. We denote by $N_\pi^{kn}(u)$ the set of vertices v such that (u, v) is either an edge or a non-edge in G_π^{kn}.

4.1 A lower bound for $\Delta = m$

4.1.1 The lower-bound construction

The graph G_1 has two components. The first component is a complete bipartite graph with $\sqrt{m}$ vertices on each side, i.e, $K_{\sqrt{m},\sqrt{m}}$, and the second component is an independent set of size $n-2\sqrt{m}$. We denote by L the set of vertices $\ell_1, \ldots, \ell_{\sqrt{m}}$ on the left-hand side of the bipartite component and by R the set of vertices $r_1, \ldots, r_{\sqrt{m}}$ on its right-hand side. The graphs in the family $\mathcal{G}_2$ have the same basic structure with a few modifications. We first choose for each graph a perfect matching M^C between the two sides R and L and remove the edges in M^C from the graph. We refer to the removed matching as the "red matching" and its pairs as "crossing non-edges" or "red pairs". Now, we add two perfect matching from L to L and from R to R, denoted M^L and M^R respectively. We refer to these matchings as the blue matchings and their edges as "non-crossing edges" or "blue pairs". Thus for each choice of three perfect matchings M^C, M^L and M^R as defined above, we have a corresponding graph in $\mathcal{G}_2$.

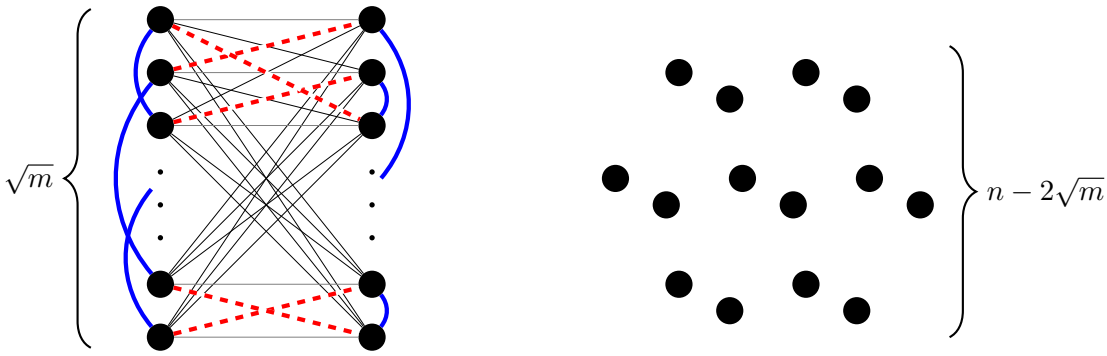


Figure 5: An illustration of the family $\mathcal{G}_2$ for $\Delta = m$.

4.1.2 Definition of the processes P_1 and P_2

In what follows we describe two random processes, P_1 and P_2, which interact with an arbitrary algorithm ALG. The process P_1 answers ALG's queries consistently with G_1. The process P_2 answers ALG's queries while constructing a uniformly selected random graph from $\mathcal{G}_2$. We assume without loss of generality that ALG does not ask queries whose answers can be derived from its

knowledge graph, since such queries give it no new information. For example, ALG does not ask a pair query about a pair of vertices that are already known to be connected by an edge due to a neighbor query. Also, we assume ALG knows in advance which vertices belong to L and which to to R, so that ALG need not query vertices in the independent set. Since the graphs in $\mathcal{G}_2$ differ from G_1 only in the edges of the subgraph induced by $L \cup R$, we think of G_1 and graphs in $\mathcal{G}_2$ as consisting only of this subgraph. Finally, since in our constructions all the vertices in $L \cup R$ have the same degree of $\sqrt{m}$, we assume that no degree queries are performed.

For every, Q, every $t \leq Q$ and every query-answer history π of length $t-1$ the process P_1 answers the t^{th} query of the algorithm consistently with G_1. Namely:

- For a pair query $q_t = (u, v)$ if the pair (u, v) is a crossing pair in G_1, then the process replies 1, and otherwise it replies 0.
- For a random new-neighbor query $q_t = u$ the process answers with a random neighbor of u that has yet been observed by the algorithm. That is, for every vertex v such that $v \in \Gamma(u) \setminus \Gamma_{\pi}^{kn}(u)$ the process replies $a_t = v$ with probability $1/(\sqrt{m} - d_{\pi}^{kn}(u))$.

The process P_2 is defined as follows:

- For a query-answer history π we denote by $\mathcal{G}_2(\pi) \subset \mathcal{G}_2$ the subset of graphs in $\mathcal{G}_2$ that are consistent with π.
- For every $t \leq Q$ and every query-answer history π of length $t-1$, the process P_2 answers the t^{th} query as follows.
  1. If the t^{th} query is a pair query $q_t = (u, v)$, then the answer a_t is selected as follows. We let $a_t = 1$ with probability
  $$\frac{|\mathcal{G}_2(\pi \circ (q_t, 1))|}{|\mathcal{G}_2(\pi)|},$$
  and let $a_t = 0$ with probability
  $$\frac{|\mathcal{G}_2(\pi \circ (q_t, 0))|}{|\mathcal{G}_2(\pi)|},$$
  where by $\circ$ we denote concatenation.
  2. If the t^{th} query is a random new-neighbor query $q_t = u_t$, then for every vertex $v \in V$ such that the pair (u_t, v) is not an edge in G_{π}^{kn}, the process answers $a_t = v$ with probability
  $$\frac{|\mathcal{G}_2(\pi \circ (q_t, v))|}{|\mathcal{G}_2(\pi)|} \cdot \frac{1}{\sqrt{m} - d_{\pi}^{kn}(u_t)}.$$
- After all queries are answered (i.e., after Q queries), uniformly choose a random graph G from $\mathcal{G}_2(\pi)$.

For a query-answer history π of length Q we denote by $\pi^{\leq t}$ the length t prefix of π and by $\pi^{\geq t}$ the $Q - t + 1$ suffix of π.

Lemma 4.1.1 *For every algorithm ALG, the process P_2, when interacting with ALG, answers ALG's queries according to a uniformly generated graph G in $\mathcal{G}_2$.*

Proof: For a specific algorithm ALG and a fixed setting r of its random coins, we denote by ALG_r the corresponding deterministic (possibly adaptive) algorithm. Consider a specific graph $G \in \mathcal{G}_2$, and let $\Pi^Q(\mathrm{ALG}_r, G)$ denote all the query-answer history of length Q that are consistent with the algorithm ALG_r and G. Since ALG_r and G are fixed throughout the proof, we shall use the shorthand Π^Q for $\Pi^Q(\mathrm{ALG}_r, G)$. For any $\pi \in \Pi^Q$, let the t^{th} query in π be denoted by $q_t(\pi)$ and the t^{th} answer by $a_t(\pi)$. The probability that G is generated by an interaction of length Q between ALG_r and the process P_2 is

$$\sum_{\pi \in \Pi^Q} \Pr_{P_2,\mathrm{ALG}_r}[\pi] \cdot \frac{1}{|\mathcal{G}_2(\pi)|} = \sum_{\pi \in \Pi^Q} \prod_{t=1}^{Q} \Pr_{P_2}[a_t(\pi) \,|\, \pi^{\leq t-1}, q_t(\pi)] \cdot \frac{1}{|\mathcal{G}_2(\pi)|} \,. \tag{76}$$

Consider first a query-answer history $\pi \in \Pi^Q$ such that $\pi = ((q_1, a_1), \ldots, (q_t, a_t))$ contains only pair queries. For every pair query q_t there is only one possible answer that is consistent with G. We denote this answer by $a_G(q_t)$. The probability that π is generated by the interaction between ALG_r and the process P_2 is

$$\begin{aligned}\Pr_{P_2}[\pi] &= \prod_{t=1}^{Q} \Pr_{P_2}[a_G(q_t) \,|\, \pi^{\leq t-1}, q_t] \\ &= \frac{\left|\mathcal{G}_2\big((q_1, a_G(q_1))\big)\right|}{|\mathcal{G}_2|} \cdot \frac{\left|\mathcal{G}_2\big((q_1, a_G(q_1)), (q_2, a_G(q_2))\big)\right|}{\left|\mathcal{G}_2\big((q_1, a_G(q_1))\big)\right|} \cdot \ldots \cdot \frac{\left|\mathcal{G}_2(\pi^{\leq Q})\right|}{|\mathcal{G}_2(\pi^{\leq Q-1})|} \,.\end{aligned} \tag{77}$$

We got a telescopic product that leaves us with

$$\Pr_{P_2,\mathrm{ALG}_r}[\pi] = \frac{\left|\mathcal{G}_2(\pi^{\leq Q})\right|}{|\mathcal{G}_2|} = \frac{|\mathcal{G}_2(\pi)|}{|\mathcal{G}_2|}.$$

Now consider a query-answer history π that consists also of random new-neighbor queries. If the t^{th} query is a random new-neighbor query $q_t = u$, then the corresponding answer a_t is some vertex v in $\Gamma_G(u) \setminus \Gamma^{kn}_{\pi^{\leq t-1}}(u)$. Recall that by the definition of the processes, given the prefix $\pi^{\leq t-1}$ and the query q_t, the probability that the process P_2 replied with v is

$$\Pr_{P_2}\left[v \,|\, \pi^{\leq t-1}, q_t\right] = \frac{|\mathcal{G}_2(\pi^{\leq t-1} \circ (q_t, v))|}{|\mathcal{G}_2(\pi^{\leq t-1})|} \cdot \frac{1}{\sqrt{m} - d^{kn}_{\pi^{\leq t-1}}(u)}.$$

Therefore, the expression for the probability $\Pr_{P_2,\mathrm{ALG}_r}[\pi]$ looks similar to the one in Equation (77), except that for queries q_t that are random new-neighbor queries there is an additional multiplicative factor of $1/(\sqrt{m} - d^{kn}_{\pi^{\leq t-1}}(u))$. As in Equation (77) all the terms will cancel each other except for the $1/(\sqrt{m} - d^{kn}_{\pi^{\leq t-1}}(u))$ terms in each random new-neighbor query and the term $\frac{|\mathcal{G}_2(\pi)|}{|\mathcal{G}_2|}$. Hence, for every query-answer history π of length Q we define the following functions:

$$\alpha_t(\pi) = \begin{cases} 1 & \text{if } q_t(\pi) \text{ is a pair query} \\ 1/\left(\sqrt{m} - d^{kn}_{\pi^{\leq t-1}}(u)\right) & \text{if } q_t(\pi) = u \text{ is a random new-neighbor query} \end{cases},$$

and

$$\alpha(\pi) = \prod_{t=1}^{Q} \alpha_t(\pi).$$

It follows from the above discussion that

$$\Pr_{P_2,ALG_r}[\pi] = \prod_{t=1}^{Q} \Pr_{P_2}[a_t \mid \pi^{\leq t-1}, q_t] \cdot \frac{|\mathcal{G}_2(\pi)|}{|\mathcal{G}_2|} = \alpha(\pi) \cdot \frac{|\mathcal{G}_2(\pi)|}{|\mathcal{G}_2|} \ .$$

By Equation (76), the probability that G is generated by ALG_r and P_2 is

$$\sum_{\pi \in \Pi^Q} \Pr_{P_2,\mathrm{ALG}_r}[\pi] \cdot \frac{1}{|\mathcal{G}_2(\pi)|} = \sum_{\pi \in \Pi Q} \alpha(\pi) \cdot \frac{|\mathcal{G}_2(\pi)|}{|\mathcal{G}_2|} \cdot \frac{1}{|\mathcal{G}_2(\pi)|} = \frac{1}{|\mathcal{G}_2|} \cdot \sum_{\pi \in \Pi^Q} \alpha(\pi) \ . \tag{78}$$

To conclude the proof we show that

$$\sum_{\pi \in \Pi^Q} \alpha(\pi) = 1 \ , \tag{79}$$

implying that the probability that G is generated is $1/|\mathcal{G}_2|$.

We prove the above by induction on the length ℓ of the history π. Observe that since ALG_r is deterministic, for any query-answer history π of length $t-1$, its next query, q_t, is uniquely determined. This implies that for every ℓ, all query-answer histories in Π^ℓ start with the same query q_1, and that if a subset of histories in Π^ℓ agree on a common prefix of queries and answers, then they also agree on the next query.

For the base of the induction, $\ell = 1$, let π be a query-answer history of length 1. If $q_1(\pi) = q_1$ is a pair query, then $\alpha(\pi) = \alpha_1(\pi) = 1$. Since there is only one answer to q_1 that is consistent with the graph G, it holds that $|\Pi^1| = 1$. Hence, we get

$$\sum_{\pi \in \Pi^1} \alpha(\pi) = \sum_{\pi = (q_1, a_G(q_1))} \alpha_1(\pi) = 1.$$

If $q_1 = u$ is a random new-neighbor query, then there are $\sqrt{m}$ answers that are consistent with the graph G, and every answer is selected with probability $1/(\sqrt{m} - d^{kn}_{\pi^{\leq t-1}}) = 1/\sqrt{m}$. Therefore, $\Pi^1 = \{(q_1, v) \mid v \in \Gamma_G(u)\}$ and

$$\sum_{\pi \in \Pi^1} \alpha(\pi) = \sum_{v \in \Gamma_G(u)} \frac{1}{\sqrt{m}} = 1.$$

This concludes the proof for the base of the induction, and we turn to the induction step.

Assume that Equation (79) holds for query-answer histories of length $\ell - 1$ where $1 \leq \ell - 1 < Q$, and consider histories of length $\ell \leq Q$. We partition the set $\Pi^{\ell-1}$ into two subsets as follows. For a query-answer history $\pi' \in \Pi^{\ell-1}$, let $q(\pi') = \mathrm{ALG}_r(\pi')$ denote the query asked by ALG_r given the query-answer history π'. The subset $\Pi^{\ell-1}_{pq}$ contains those histories $\pi' \in \Pi^{\ell-1}$ for which the query $q(\pi')$ is a pair query on some pair (u, v) and the subset $\Pi^{\ell-1}_{nq}$ contains those histories $\pi' \in \Pi^{\ell-1}$ for which the query $q(\pi')$ is a random new-neighbor query on some vertex u.

Observe that for each $\pi' \in \Pi^{\ell-1}_{pq}$ there is one possible answer to $q(\pi')$ that is consistent with G, denoted $a_G(q(\pi'))$. It follows that for each $\pi' \in \Pi^{\ell-1}_{pq}$ there is a single query-answer history $\pi \in \Pi^\ell$ such that $\pi^{\leq \ell-1} = \pi'$, namely, $\pi = \pi' \circ (q(\pi'), a_G(q(\pi')))$, and $\alpha_\ell(\pi) = 1$. For each $\pi' \in \Pi^{\ell-1}_{nq}$ there are $\sqrt{m} - d^{kn}_{\pi'}(q(\pi'))$ possible answers that are consistent with G, one for each neighbor of $q(\pi')$ that is not in $\Gamma^{kn}_{\pi'}(q(\pi'))$. It follows that for each $\pi' \in \Pi^{\ell-1}_{nq}$ there are $\sqrt{m} - d^{kn}_{\pi'}(q(\pi'))$ query-answer histories $\pi \in \Pi^\ell$ such that $\pi^{\leq \ell-1} = \pi'$. Each of them is of the form $\pi = \pi' \circ (q(\pi'), v)$ for

$v \in \Gamma_G(q(\pi')) \setminus \Gamma^{kn}_{\pi'}(q(\pi'))$, and for each $\alpha_\ell(\pi) = 1/(\sqrt{m} - d^{kn}_{\pi'}(q(\pi')))$. Therefore,

$$\begin{aligned}
\sum_{\pi \in \Pi^\ell} \alpha(\pi) &= \sum_{\pi' \in \Pi^{\ell-1}_{pq}} \alpha(\pi') \cdot \alpha_\ell\big(\pi' \circ (q(\pi'), a_G(q(\pi')))\big) \\
&\quad + \sum_{\pi' \in \Pi^{\ell-1}_{nq}} \alpha(\pi') \cdot \sum_{v \in \Gamma_G(q(\pi')) \setminus \Gamma^{kn}_{\pi'}(q(\pi'))} \alpha_\ell\big(\pi' \circ (q(\pi'), v)\big)\big) \\
&= \sum_{\pi' \in \Pi^{\ell-1}_{pq}} \alpha(\pi') + \sum_{\pi' \in \Pi^{\ell-1}_{nq}} \alpha(\pi') \cdot \sum_{v \in \Gamma_G(q(\pi')) \setminus \Gamma^{kn}_{\pi'}(q(\pi'))} \frac{1}{\sqrt{m} - d^{kn}_{\pi'}(q(\pi'))} \\
&= \sum_{\pi' \in \Pi^{\ell-1}} \alpha(\pi') \;=\; 1 \qquad (80)
\end{aligned}$$

where in the last equality we used the induction hypothesis. This concludes the proof that Equation (79) holds for all query-answer histories π of length at most Q. Combining Equation (79) with Equation (78) completes the proof of the lemma. ■

For a fixed algorithm ALG that performs Q queries, and for $b \in \{1, 2\}$, let $\mathcal{D}^b_{\text{ALG}}$ denote the distribution on query-answers histories of length Q induced by the interaction between ALG and P_b. We shall show that for every algorithm ALG that performs at most $Q = \frac{m^{3/2}}{100\Delta}$ queries, the statistical distance between $\mathcal{D}^{\text{ALG}}_1$ and $\mathcal{D}^{\text{ALG}}_2$, denoted $d\left(\mathcal{D}^{\text{ALG}}_1, \mathcal{D}^{\text{ALG}}_2\right)$, is at most $\frac{1}{3}$. This will imply that the lower bound stated in Theorem 4.2 holds for the case that $\Delta(G) = m$. In order to obtain this bound we introduce the notion of a query-answer witness pair, defined next.

Definition 4.1.1 *We say that ALG has detected a query-answer witness pair in three cases:*

1. *If q_t is a pair query for a crossing pair $(u_t, v_t) \in L \times R$ and $a_t = 0$.*
2. *If q_t is a pair query for a non-crossing pair $(u_t, v_t) \in (L \times L) \cup (R \times R)$ and $a_t = 1$.*
3. *If $q_t = u_t$ is a random new-neighbor query and $a_t = v$ for some v such that (u_t, v) is a non-crossing pair.*

We note that the source of the difference between $\mathcal{D}^{\text{ALG}}_1$ and $\mathcal{D}^{\text{ALG}}_2$ is not only due to the probability that the query-answer history contains a witness pair (which is 0 under $\mathcal{D}^{\text{ALG}}_1$ and non-0 under $\mathcal{D}^{\text{ALG}}_2$). There is also a difference in the distribution over answers to random new neighbor queries when the answers do not result in witness pairs (in particular when we condition on the query-answer history prior to the t^{th} query). However, the analysis of witness pairs serves us also in bounding the contribution to the distance due to random new neighbor queries that do not result in a witness pairs.

Let w be a "witness function", such that for a pair query q_t on a crossing pair, $w(q_t) = 0$, and for a non-crossing pair, $w(q_t) = 1$. The probability that ALG detects a witness pair when q_t is a pair query (u_t, v_t) and π is a query-answer history of length $t - 1$, is

$$\Pr_{P_2}[w(q_t) \mid \pi] = \frac{|\mathcal{G}_2(\pi \circ (q_t, w(q_t)))|}{|\mathcal{G}_2(\pi)|} \leq \frac{|\mathcal{G}_2(\pi \circ (q_t, w(q_t)))|}{|\mathcal{G}_2(\pi \circ (q_t, \overline{w(q_t)}))|} .$$

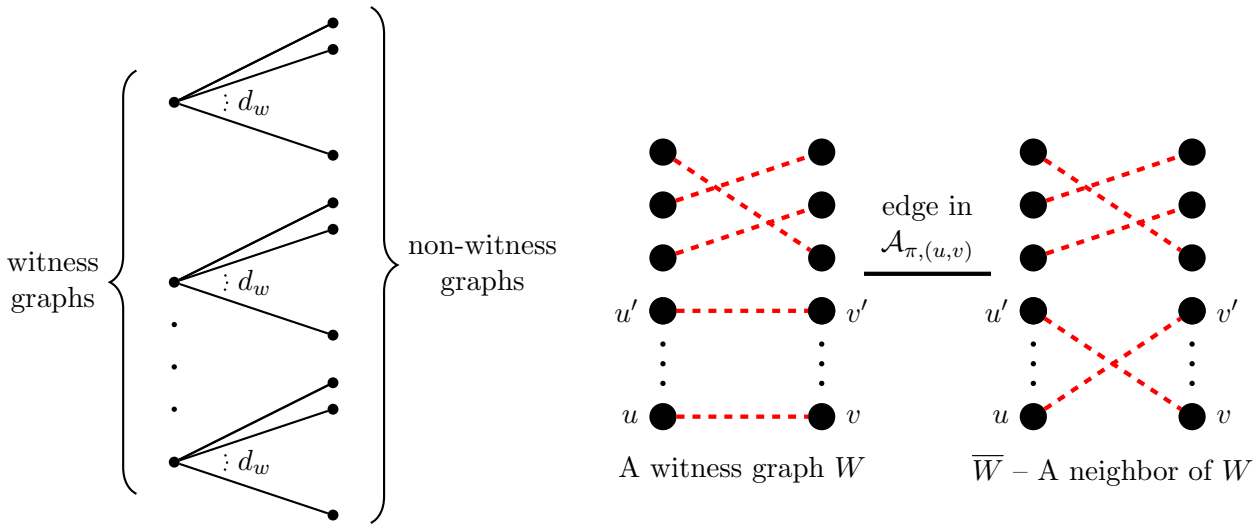
Therefore, to bound the probability that the algorithm observes a witness pair it is sufficient to bound the ratio between the number of graphs in $\mathcal{G}_2(\pi \circ (q, w(q_t)))$ and the number of graphs in $\mathcal{G}_2(\pi \circ (q, \overline{w(q_t)}))$. We do this by introducing an auxiliary graph, which is defined next.

4.1.3 The auxiliary graph for $\Delta = m$

For every $t \leq Q$, every query-answer history π of length $t-1$ for which π is consistent with G_1 (that is, no witness pair has yet been detected), and every pair (u, v), we consider a bipartite auxiliary graph $\mathcal{A}_{\pi,(u,v)}$. On one side of $\mathcal{A}_{\pi,(u,v)}$ we have a node for every graph in $\mathcal{G}_2(\pi)$ for which the pair (u, v) is a witness pair. We refer to these nodes as witness graphs. On the other side of the auxiliary graph, we place a node for every graph in $\mathcal{G}_2(\pi)$ for which the pair is not a witness. We refer to these nodes as non-witness graphs. We put an edge in the auxiliary graph between a witness graph W and a non-witness graph $\overline{W}$ if the pair (u, v) is a crossing (non-crossing) pair and the two graphs are identical except that their red (blue) matchings differ on exactly two pairs – (u, v) and one additional pair. In other words, $\overline{W}$ can be obtained from W be performing a *switch* operation, as defined next.

Definition 4.1.2 *We define a switch between pairs in a matching in the following manner. Let (u, v) and (u', v') be two matched pairs in a matching M. A switch between (u, v) and (u', v') means removing the edges (u, v) and (u', v') from M and adding to it the edges (u, v') and (u', v).*

Note that the switch process maintains the cardinality of the matching. We denote by $d_w(\mathcal{A}_{\pi,(u,v)})$ the minimal degree of any witness graph in $\mathcal{A}_{\pi,(u,v)}$, and by $d_{nw}(\mathcal{A}_{\pi,(u,v)})$ the maximal degree of the non-witness graphs. See Figure 6 for an illustration.



(a) The auxiliary graph with witness nodes on the left and non-witness nodes on the right.

(b) An illustration of two neighbors in the auxiliary graph for $\Delta = m$.

Figure 6

Lemma 4.1.2 *Let $\Delta = m$ and $Q = \frac{m^{3/2}}{100\Delta}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and every pair (u, v),*

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} \leq \frac{2}{\sqrt{m}} = \frac{2\Delta}{m^{3/2}}.$$

Proof: Recall that the graphs in $\mathcal{G}_2$ are as defined in Subsection 4.1.1 and illustrated in Figure 5. In the following we consider crossing pairs, as the proof for non-crossing pairs is almost identical. Recall that a crossing pair is a pair (u, v) such that $u \in L$ and $v \in R$ or vise versa. A witness graph W with respect to the pair (u, v) is a graph in which (u, v) is a red pair, i.e., $(u, v) \in M^C$. There is an edge from W to every non-witness graph $\overline{W} \in \mathcal{G}_2(\pi)$ such that $M^C(W)$ and $M^C(\overline{W})$ differ exactly on (u, v) and one additional edge.

Every red pair $(u', v') \in M^C(W)$ creates a potential non-witness graph $\overline{W}_{(u',v')}$ when switched with (u, v) (as defined in Definition 4.1.2). However, not all of the these non-witness graphs are in $\mathcal{G}_2(\pi)$. If u' is a neighbor of v in the knowledge graph G^{kn}_π, i.e., $u' \in \Gamma^{kn}_\pi(v)$, then $\overline{W}_{(u',v')}$ is not consistent with the knowledge graph, and therefore $\overline{W}_{(u',v')} \notin \mathcal{G}_2(\pi)$. This is also the case for a pair (u', v') such that $v' \in \Gamma^{kn}_\pi(u)$. Therefore, only pairs $(u', v') \in M^C$ such that $u' \notin \Gamma^{kn}_\pi(v)$ and $v' \notin \Gamma^{kn}_\pi(u)$ produce a non-witness graph $\overline{W}_{(u',v')} \in \mathcal{G}_2(\pi)$ when switched with (u, v). We refer to these pairs as consistent pairs. Since $t \leq \frac{\sqrt{m}}{100}$, both u and v each have at most $\frac{m}{100}$ neighbors in the knowledge graph, implying that out of the $\sqrt{m} - 1$ potential pairs, the number of consistent pairs is at least

$$\sqrt{m} - 1 - d^{kn}_\pi(u) - d^{kn}_\pi(v) \geq \sqrt{m} - 1 - 2 \cdot \frac{\sqrt{m}}{100} \geq \frac{1}{2}\sqrt{m}.$$

Therefore, the degree of every witness graph $W \in \mathcal{A}_{\pi,(u,v)}$ is at least $\frac{1}{2}\sqrt{m}$, implying that $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{2}\sqrt{m}$.

In order to prove that $d_{nw}(\mathcal{A}_{\pi,(u,v)}) = 1$, consider a non-witness graph $\overline{W}$. Since $\overline{W}$ is a non-witness graph, the pair (u, v) is not a red pair. This implies that u is matched to some vertex $v' \in R$, and v is matched to some vertex $u' \in L$. That is, $(u, v'), (v, u') \in M^C$. By the construction of the edges in the auxiliary graph, every neighbor W of $\overline{W}$ can be obtained by a single switch between two red pairs in the red matching. The only possibility to switch two pairs in $M^C(\overline{W})$ and obtain a matching in which (u, v) is a red pair is to switch the pairs (u, v') and (v, u'). Hence, every non-witness graph $\overline{W}$ has at most one neighbor.

We showed that $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{2}\sqrt{m}$ and that $d_{nw}(\mathcal{A}_{\pi,(u,v)}) \leq 1$, implying

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} \leq \frac{2}{\sqrt{m}} = \frac{2\Delta}{m^{3/2}},$$

and the proof is complete. ■

4.1.4 Statistical distance

For a query-answer history π of length $t-1$ and a query q_t, let $Ans(\pi, q_t)$ denote the set of possible answers to the query q_t that are consistent with π. Namely, if q_t is a pair query (for a pair that does not belong to the knowledge graph G^{kn}_π), then $Ans(\pi, q_t) = \{0, 1\}$, and if q_t is a random new-neighbor query, then $Ans(\pi, q_t)$ consists of all vertices except those in N^{kn}_π.

Lemma 4.1.3 *Let $\Delta = m$ and $Q = \frac{m^{3/2}}{100\Delta}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and for every query q_t:*

$$\sum_{a \in Ans(\pi,q_t)} \left| \Pr_{P_1}[a \mid \pi, q_t] - \Pr_{P_2}[a \mid \pi, q_t] \right| \leq \frac{12}{\sqrt{m}} = \frac{12\Delta}{m^{3/2}}.$$

Proof: We prove the lemma separately for each type of query.

1. We start with a crossing pair query (u_t, v_t). In this case the witnesses are red pairs. Namely, our witness graphs for this case are all the graphs in $\mathcal{G}_2(\pi \circ (q_t, 0))$, and the non-witness graphs are all the graphs in $\mathcal{G}_2(\pi \circ (q_t, 1))$. By the construction of the auxiliary graph

$$|\mathcal{G}_2(\pi \circ (q_t, 0))| \cdot d_w(\mathcal{A}_{\pi,(u,v)}) \leq |\mathcal{G}_2(\pi \circ (q_t, 1))| \cdot d_{nw}(\mathcal{A}_{\pi,(u,v)}).$$

   This, together with Lemma 4.1.2, implies

$$\frac{|\mathcal{G}_2(\pi \circ (q_t, 0))|}{|\mathcal{G}_2(\pi)|} \leq \frac{|\mathcal{G}_2(\pi \circ (q_t, 0))|}{|\mathcal{G}_2(\pi \circ (q_t, 1))|} \leq \frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} = \frac{2}{\sqrt{m}} = \frac{2\Delta}{m^{3/2}} .$$

   For a pair query q_t, the set of possible answers $Ans(\pi, q_t)$ is $\{0, 1\}$. Therefore,

$$\begin{aligned}
&\sum_{a \in \{0,1\}} \left| \Pr_{P_1}[a \mid \pi, q_t] - \Pr_{P_2}[a \mid \pi, q_t] \right| \\
&\quad = \left| \Pr_{P_1}[0 \mid \pi, q_t] - \Pr_{P_2}[0 \mid \pi, q_t] \right| + \left| \Pr_{P_1}[1 \mid \pi, q_t] - \Pr_{P_2}[1 \mid \pi, q_t] \right| \\
&\quad = \frac{2\Delta}{m^{3/2}} + 1 - \left(1 - \frac{2\Delta}{m^{3/2}}\right) = \frac{4\Delta}{m^{3/2}} = \frac{4}{\sqrt{m}} .
\end{aligned} \tag{81}$$

2. For a non-crossing pair query $q_t = (u, v)$ our witness graphs are graphs that contain q_t as a blue pair, i.e., graphs from $\mathcal{G}_2(\pi, (q_t, 1))$, and our non-witness graphs are graphs in which no blue pair had been queried, i.e., graphs from $\mathcal{G}_2(\pi, (q_t, 0))$. From Lemma 4.1.2 we get that for a non-crossing pair query q_t:

$$\frac{|\mathcal{G}_2(\pi \circ (q_t, 1))|}{|\mathcal{G}_2(\pi)|} \leq \frac{|\mathcal{G}_2(\pi \circ (q_t, 1))|}{|\mathcal{G}_2(\pi \circ (q_t, 0))|} \leq \frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} = \frac{2\Delta}{m^{3/2}} = \frac{2}{\sqrt{m}} .$$

   Therefore,

$$\begin{aligned}
&\sum_{a \in \{0,1\}} \left| \Pr_{P_1}[a \mid \pi, q_t] - \Pr_{P_2}[a \mid \pi, q_t] \right| \\
&\quad = \left| \Pr_{P_1}[0 \mid \pi, q_t] - \Pr_{P_2}[0 \mid \pi, q_t] \right| + \left| \Pr_{P_1}[1 \mid \pi, q_t] - \Pr_{P_2}[1 \mid \pi, q_t] \right| \\
&\quad = 1 - \left(1 - \frac{2\Delta}{m^{3/2}}\right) + \frac{2\Delta}{m^{3/2}} = \frac{4\Delta}{m^{3/2}} = \frac{4}{\sqrt{m}} .
\end{aligned} \tag{82}$$

3. For a new-neighbor query $q_t = u_t$, the set of possible answers $Ans(\pi, q_t)$ is the set of all the vertices in the graph. Therefore,

$$\begin{aligned}
&\sum_{a \in Ans(\pi, q_t)} \left| \Pr_{P_1}[a \mid \pi, q_t] - \Pr_{P_2}[a \mid \pi, q_t] \right| \\
&= \sum_{v \in R} \left| \Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t] \right| + \sum_{v \in L} \left| \Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t] \right| .
\end{aligned}$$

Recall that for a vertex $v \in \Gamma_\pi^{kn}(u)$, $\Pr_{P_1}[v \mid \pi, q_t] = \Pr_{P_2}[v \mid \pi, q_t] = 0$. Therefore, it suffices to consider only vertices v such that $v \notin \Gamma_\pi^{kn}(u)$. Assume without loss of generality that $u \in L$, and consider a vertex $v \in R, v \notin \Gamma_\pi^{kn}(u)$. Since for every $v \in R$ we have that $(u_t, v) \in E(G_1)$, by the definition of P_1,

$$\Pr_{P_1}[v \mid \pi, q_t] = \frac{1}{\sqrt{m} - d_\pi^{kn}(u_t)} \,. \tag{83}$$

Now consider the process P_2. By its definition,

$$\begin{aligned}
\Pr_{P_2}[v \mid \pi, q_t] &= \frac{\mathcal{G}_2\left(\pi \circ (q_t, v)\right)}{\mathcal{G}_2(\pi)} \cdot \frac{1}{\sqrt{m} - d_\pi^{kn}(u)} \\
&= \frac{\mathcal{G}_2\left(\pi \circ ((u, v), 1)\right)}{\mathcal{G}_2(\pi)} \cdot \frac{1}{\sqrt{m} - d_\pi^{kn}(u)} \\
&= \left(1 - \frac{\mathcal{G}_2\left(\pi \circ ((u, v), 0)\right)}{\mathcal{G}_2(\pi)}\right) \cdot \frac{1}{\sqrt{m} - d_\pi^{kn}(u)} \,.
\end{aligned}$$

By the first item in the proof, for any crossing pair $q_t = (u, v)$,

$$\frac{\mathcal{G}_2(\pi \circ (q_t, 0))}{\mathcal{G}_2(\pi)} = \frac{4\Delta}{m^{3/2}} = \frac{4}{\sqrt{m}} \,,$$

and it follows that

$$\Pr_{P_2}[v \mid \pi, q_t] = \left(1 - \frac{4\Delta}{m^{3/2}}\right) \cdot \frac{1}{\sqrt{m} - d_\pi^{kn}(u)} \,. \tag{84}$$

By Equations (83) and (84), we get that for every $v \in R$ such that $v \notin \Gamma_\pi^{kn}(u)$,

$$\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| = \frac{4\Delta/m^{3/2}}{\sqrt{m} - d_\pi^{kn}(u)}. \tag{85}$$

Therefore,

$$\begin{aligned}
\sum_{v \in R}\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| &= \sum_{v \in R, v \notin \Gamma_\pi^{kn}(u)}\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| \\
&= \left(\sqrt{m} - d_\pi^{kn}(u)\right) \cdot \frac{4\Delta/m^{3/2}}{\sqrt{m} - d_\pi^{kn}(u)} = \frac{4\Delta}{m^{3/2}} = \frac{4}{\sqrt{m}} \,.
\end{aligned} \tag{86}$$

Now consider a vertex $v \in L$. Observe that for every $v \in L$, it holds that $v \notin \Gamma_\pi^{kn}(u)$ since otherwise π is not consistent with G_1. For the same reason,

$$\Pr_{P_1}[v \mid \pi, q_t] = 0 \,. \tag{87}$$

As for P_2, as before,

$$\Pr_{P_2}[v \mid \pi, q_t] = \frac{\mathcal{G}_2(\pi, (u_t, v))}{\mathcal{G}_2(\pi)} \cdot \frac{1}{\sqrt{m} - d_\pi^{kn}(u_t)} \,.$$

By the second item of the claim, since for every $v \in L$, (u_t, v) is a non-crossing pair, we have that

$$\frac{|\mathcal{G}_2(\pi,(u_t,v))|}{|\mathcal{G}_2(\pi)|} = \frac{4\Delta}{m^{3/2}} = \frac{4}{\sqrt{m}} \,. \tag{88}$$

Combining Equations (87) and (88) we get that for every $v \in L$

$$\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| = \frac{4\Delta/m^{3/2}}{\sqrt{m} - d^{kn}_{\pi}(u)}.$$

Since $Q = \frac{m^{3/2}}{100\Delta} = \frac{\sqrt{m}}{100}$, for every $t \leq Q$, $d^{kn}_{\pi}(u) < \frac{1}{2}\sqrt{m}$, and it follows that $\frac{\sqrt{m}-1}{\sqrt{m}-d^{kn}(u)}$ is bounded by 2. Hence,

$$\begin{aligned}\sum_{v\in L}\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| &= (\sqrt{m}-1)\cdot\frac{4\Delta/m^{3/2}}{\sqrt{m} - d^{kn}_{\pi}(u)} \\ &= \frac{8\Delta}{m^{3/2}} = \frac{8}{\sqrt{m}} \,. \end{aligned} \tag{89}$$

By Equations (86) and (89) we get

$$\begin{aligned}&\sum_{v\in R}\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| + \sum_{v\in L}\left|\Pr_{P_1}[v \mid \pi, q_t] - \Pr_{P_2}[v \mid \pi, q_t]\right| \\ &\quad = \frac{12\Delta}{m^{3/2}} = \frac{12}{\sqrt{m}}. \end{aligned} \tag{90}$$

This completes the proof. ■

Recall that $\mathcal{D}^{\mathrm{ALG}}_b$, $b \in \{1,2\}$, denotes the distribution on query-answer histories of length Q, induced by the interaction of ALG and P_b. We will show that the two distributions are indistinguishable for Q that is sufficiently small.

Lemma 4.1.4 *Let $\Delta = m$. For every algorithm ALG that asks at most $Q = \frac{m^{3/2}}{100\Delta}$ queries, the statistical distance between $\mathcal{D}^{ALG}_1$ and $\mathcal{D}^{ALG}_2$ is at most $\frac{1}{3}$.*

Proof: Consider the following hybrid distribution. Let $\mathcal{D}^{\mathrm{ALG}}_{1,t}$ be the distribution over query-answer histories of length Q, where in the length t prefix ALG is answered by the process P_1 and in the length $Q-t$ suffix ALG is answered by the process P_2. Observe that $\mathcal{D}^{\mathrm{ALG}}_{1,Q} = \mathcal{D}^{\mathrm{ALG}}_1$ and that $\mathcal{D}^{\mathrm{ALG}}_{1,0} = \mathcal{D}^{\mathrm{ALG}}_2$. Let $\pi = (\pi_1, \pi_2, \ldots, \pi_\ell)$ denote a query-answer history of length ℓ. By the triangle inequality

$$d(\mathcal{D}^{\mathrm{ALG}}_1, \mathcal{D}^{\mathrm{ALG}}_2) \leq \sum_{t=0}^{Q-1} d(\mathcal{D}^{\mathrm{ALG}}_{1,t+1}, \mathcal{D}^{\mathrm{ALG}}_{1,t}) \,.$$

It thus remains to bound $d(\mathcal{D}^{\mathrm{ALG}}_{1,t+1}, \mathcal{D}^{\mathrm{ALG}}_{1,t}) = \frac{1}{2}\sum_{\pi}\left|\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t+1}}[\pi] - \Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t}}[\pi]\right|$ for every $0 \leq t \leq Q-1$. Let $\mathcal{Q}$ denote the set of all possible queries.

$$\sum_{\pi}\left|\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t}}[\pi]\right|$$
$$=\sum_{\pi_1,\ldots,\pi_{t-1}}\Pr_{P_1,\mathrm{ALG}}[\pi_1,\ldots,\pi_{t-1}]\cdot\sum_{q\in\mathcal{Q}}\Pr_{\mathrm{ALG}}[q\mid\pi_1,\ldots,\pi_{t-1}]$$
$$\cdot\sum_{a\in Ans((\pi_1,\ldots,\pi_{t-1}),q)}\left|\Pr_{P_1}[a\mid\pi_1,\ldots,\pi_{t-1},q]-\Pr_{P_2}[a\mid\pi_1,\ldots,\pi_{t-1},q]\right|$$
$$\cdot\sum_{\pi_{t+1},\ldots,\pi_Q}\Pr_{P_2,\mathrm{ALG}}[\pi_{t+1},\ldots,\pi_Q\mid\pi_1,\ldots,\pi_{t-1},(q,a)]\ .$$

By Lemma 4.1.3, for every $1\le t\le Q-1$, and every $\pi_1,\ldots,\pi_{t-1}$ and q,

$$\sum_{a\in Ans((\pi_1,\ldots,\pi_{t-1}),q)}\left|\Pr_{P_1}[a\mid\pi_1,\ldots,\pi_{t-1},q]-\Pr_{P_2}[a\mid\pi_1,\ldots,\pi_{t-1},q]\right|\le\frac{12\Delta}{m^{3/2}}\ .$$

We also have that for every pair (q,a),

$$\sum_{\pi_{t+1},\ldots,\pi_Q}\Pr_{P_2,\mathrm{ALG}}[\pi_{t+1},\ldots,\pi_Q\mid\pi_1,\ldots,\pi_{t-1},(q,a)]=1\ .$$

Therefore,

$$\sum_{\pi}\left|\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t}}[\pi]\right|$$
$$\le\sum_{\pi_1,\ldots,\pi_{t-1}}\Pr_{P_1,\mathrm{ALG}}[\pi_1,\ldots,\pi_{t-1}]\sum_{q\in\mathcal{Q}}\Pr_{\mathrm{ALG}}[q\mid\pi_1,\ldots,\pi_{t-1}]\cdot\frac{12\Delta}{m^{3/2}}=\frac{12\Delta}{m^{3/2}}\ .$$

Therefore, for $Q=\frac{\sqrt{m}}{100}$

$$d(\mathcal{D}^{\mathrm{ALG}}_1,\mathcal{D}^{\mathrm{ALG}}_2)=\frac{1}{2}\sum_{\pi}\sum_{t=1}^{Q-1}\left|\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\mathcal{D}^{\mathrm{ALG}}_{1,t}}[\pi]\right|\le\frac{1}{2}\cdot Q\cdot\frac{12\Delta}{m^{3/2}}\le\frac{1}{3},$$

and the proof is complete. ■

In the next subsection we turn to prove the theorem for the cases where $m<\Delta\le\frac{m^{3/2}}{8}$, and for the case where $\sqrt{m}\le\Delta\le\frac{m}{4}$. We start with the former case. The proof will follow the building blocks of the proof for $\Delta=\sqrt{m}$, where the only difference is in the description of the auxiliary graph $\mathcal{A}_{\pi,(u,v)}$ and in the proof that $\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})}\le\frac{2\Delta}{m^{3/2}}=\frac{2r}{\sqrt{m}}$.

4.2 A lower bound for $m<\Delta<m^{3/2}$

Let $\Delta=r\cdot m$ for an integer r such that $1<r\le\frac{1}{8}\sqrt{m}$. It is sufficient for our needs to consider only values of Δ for which r is an integer. The proof of the lower bound for this case is a fairly simple extension of the proof for the case of $\Delta=m$, that is, $r=1$. We next describe the modifications we make in the construction of $\mathcal{G}_2$.

4.2.1 The lower-bound construction

Let G_1 be as defined in Subsection 4.1.1. The construction of $\mathcal{G}_2$ for $\Delta = r \cdot m$ can be thought of as repeating the construction of $\mathcal{G}_2$ for $\Delta = m$ (as described in Subsection 4.1.1) r times. We again start with a complete bipartite graph $K_{\sqrt{m},\sqrt{m}}$ and an independent set of size $n - 2\sqrt{m}$. For each graph $G \in \mathcal{G}_2$ we select r perfect matchings between the two sides R and L and remove these edges from the graph. We denote the r perfect matchings by $M_1^C, \ldots, M_r^C$ and refer to them as the red matchings. We require that each two perfect matchings M_i^C and M_j^C do not have any shared edges. That is, for every i and for every j, for every $(u, v) \in M_i^C$ it holds that $(u, v) \notin M_j^C$. In order to maintain the degrees of the vertices, we next select r perfect matchings for each side of the bipartite graph (L to L and R to R). We denote these matchings by $M_1^R, ..., M_r^R$ and $M_1^L, ..., M_r^L$ respectively. Again we require that no two matchings share an edge. We refer to these matchings as the blue matchings and their edges as blue pairs. Each such choice of $3r$ matchings defines a graph in $\mathcal{G}_2$.

4.2.2 The processes P_1 and P_2

The definition of the processes P_1 and P_2 is the same as in Subsection 4.1.2 (using the modified definition of $\mathcal{G}_2$), and Lemma 4.1.1 holds here as well.

4.2.3 The auxiliary graph

As before, for every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and every pair (u, v), we define a bipartite auxiliary graph $\mathcal{A}_{\pi,(u,v)}$, such that on one side there is a node for every witness graph $W \in \mathcal{G}_2(\pi)$, and on the other side a node for every non-witness graph $\overline{W} \in \mathcal{G}_2(\pi)$. The witness graphs for this case are graphs in which (u, v) is a red (blue) edge in one of the red (blue) matchings. If (u, v) is a crossing pair, then for every witness graph W, $(u, v) \in M_i^C(W)$ for some $1 \leq i \leq r$. If (u, v) is a non-crossing pair, then for every witness graph W, $(u, v) \in M_i^L(W)$ or $(u, v) \in M_i^L(W)$. There is an edge from W to every graph $\overline{W}$ such that the matching that contains (u, v) in W and the corresponding matching in $\overline{W}$ differ on exactly two pairs – (u, v) and one additional pair. For example, if $(u, v) \in M_i^C(W)$, there is an edge from W to every graph $\overline{W}$ such that $M_i^C(W)$ and $M_i^C(\overline{W})$ differ on exactly (u, v) and one additional pair.

Lemma 4.2.1 *Let $\Delta = r \cdot m$ for an integer r such that $1 < r \leq \frac{\sqrt{m}}{8}$ and let $Q = \frac{m^{3/2}}{100\Delta}$. For every $t \leq Q$, every query-answer history π of length $t - 1$ such that π is consistent with G_1 and every pair (u, v),*

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} \leq \frac{2\Delta}{m^{3/2}} = \frac{2r}{\sqrt{m}}.$$

Proof: We again analyze the case in which the pair is a crossing pair (u, v), as the proof for a non-crossing pair is almost identical. We first consider the minimal degree of the witness graphs in $\mathcal{A}_{\pi,(u,v)}$. Let M_i^C be the matching to which (u, v) belongs. As before, only pairs $(u', v') \in M_i^C$ such that $u' \notin \Gamma_\pi^{kn}(u)$, $v' \notin \Gamma_\pi^{kn}(v)$ result in a non-witness graph $\overline{W} \in \mathcal{G}_2(\pi)$ when switched with (u, v). However, we have an additional constraint. Since by our construction no two red matchings share an edge, it must be that u' is not matched to v in any of the other r red matching, and similarly that u is not matched to v' in any of the other matchings. It follows that of the $(\sqrt{m} - 1 - 2 \cdot \frac{m^{3/2}}{100 \cdot r \cdot m})$

potential pairs (as in the proof of Lemma 4.1.2), we discard $2r$ additional pairs. Since $1 \leq r \leq \frac{\sqrt{m}}{8}$ we remain with $(\sqrt{m} - 1 - \frac{\sqrt{m}}{50} - \frac{1}{4}\sqrt{m}) \geq \frac{1}{2}\sqrt{m}$ potential pairs. Thus, $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{2}\sqrt{m}$.

We now turn to consider the degree of the non-witness graphs and prove that $d_{nw}(\mathcal{A}_{\pi,(u,v)}) \leq r$. Consider a non-witness graph $\overline{W}$. To prove that $\overline{W}$ has at most r neighbors it is easier to consider all the possible options to "turn" $\overline{W}$ from a non-witness graph into a witness graph. It holds that for every $j \in [r]$, $(u,v) \notin M_j^C(\overline{W})$. Therefore for every matching M_j^C, u is matched to some vertex, denoted v'_j and v is matched to some vertex, denoted u'_j. If we switch between the pairs (u,v'_j) and (v,u'_j), this results in a matching in which (u,v) is a witness pair. We again refer the reader to Figure 6b, where the illustrated matching can be thought of as the j^{th} matching. Denote the resulting graph by $W_{(u'_j,v'_j)}$. If the pair (u'_j,v'_j) has not been observed yet by the algorithm then $W_{(u'_j,v'_j)}$ is a witness graph in $\mathcal{A}_{\pi,(u,v)}$. Therefore there are at most r options to turn $\overline{W}$ into a witness graph, and $d_{nw}(\mathcal{A}_{\pi,(u,v)}) \leq r$. We showed that $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{2}\sqrt{m}$ and $d_{nw}(\mathcal{A}_{\pi,(u,v)}) \leq r$, implying

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} \leq \frac{2r}{\sqrt{m}} = \frac{2\Delta}{m^{3/2}},$$

as required. ■

4.2.4 Statistical distance

The proof of the next lemma is exactly the same as the proof of Lemma 4.1.3, except that occurrences of the term $(\Delta/m^{3/2})$ are replaced by $(r/\sqrt{m})$ instead of $(1/\sqrt{m})$, and we apply Lemma 4.2.1 instead of Lemma 4.1.2.

Lemma 4.2.2 *Let $\Delta = r \cdot m$ for an integer r such that $1 < r \leq \frac{\sqrt{m}}{8}$ and let $Q = \frac{m^{3/2}}{100\Delta}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and for every query q_t,*

$$\sum_{a \in Ans(\pi,q_t)} \left| \Pr_{P_1}[a \mid \pi, q_t] - \Pr_{P_2}[a \mid \pi, q_t] \right| = \frac{12\Delta}{m^{3/2}} = \frac{12r}{\sqrt{m}} .$$

The proof of the next lemma is same as the proof of Lemma 4.1.4 except that we replace the application of Lemma 4.1.3, by an application of Lemma 4.2.2.

Lemma 4.2.3 *Let $\Delta = r \cdot m$ for an integer r such that $1 < r \leq \frac{\sqrt{m}}{8}$. For every algorithm ALG that performs at most $Q = \frac{m^{3/2}}{100\Delta}$ queries, the statistical distance between $\mathcal{D}_1^{ALG}$ and $\mathcal{D}_2^{ALG}$ is at most $\frac{1}{3}$.*

4.3 A lower bound for $\sqrt{m} \leq \Delta \leq \frac{1}{4}m$

Similarly to the previous section, we let $\Delta = k\sqrt{m}$ and assume that k is an integer such that $1 \leq k \leq \frac{\sqrt{m}}{4}$.

4.3.1 The lower-bound construction

The construction of the graph G_1 is as defined in Subsection 4.1.1, and we modify the construction of the graphs in $\mathcal{G}_2$. As before, the basic structure of every graph is a complete bipartite graph $K_{\sqrt{m},\sqrt{m}}$

and an independent set of size $n-2\sqrt{m}$ vertices. In this case, for each graph in $\mathcal{G}_2$, we do not remove a perfect matching from the bipartite graph, but rather a matching M^C of size k. In order to keep the degrees of all vertices to be $\sqrt{m}$, we modify the way we construct the blue matchings. Let $M^C = \{(\ell_{i_1}, r_{i_1}), (\ell_{i_2}, r_{i_2}), \ldots, (\ell_{i_k}, r_{i_k})\}$ be the crossing matching. The blue matchings will be $M^L = \{(\ell_{i_1}, \ell_{i_2}), (\ell_{i_3}, \ell_{i_4}), \ldots, (\ell_{i_k-1}, \ell_{i_k})\}$ and $M^R = \{(r_{i_1}, r_{i_2}), (r_{i_3}, r_{i_4}), \ldots, (r_{i_k-1}, r_{i_k})\}$. Note that every matched pair belongs to a four-tuple $\langle \ell_{i_j}, \ell_{i_{j+1}}, r_{i_{j+1}}, r_{i_j} \rangle$ such that (ℓ_{i_j}, r_{i_j}) and $(\ell_{i_{j+1}}, r_{i_{j+1}})$ are red pairs and $(\ell_{i_j}, \ell_{i_{j+1}})$ and $(r_{i_j}, r_{i_{j+1}})$ are blue pairs. We refer to these structures as matched squares and to four-tuples $(\ell_x, \ell_y, r_z, r_w)$ such that no pair in the tuple is matched as unmatched squares. See Figure 7 for an illustration. Every graph in $\mathcal{G}_2$ is defined by its set of k four-tuples.

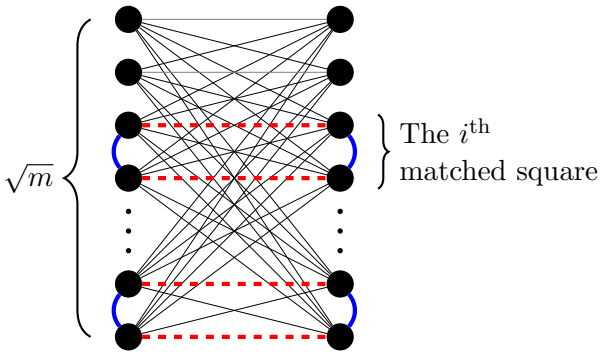


Figure 7: An illustration of the bipartite component in the family $\mathcal{G}_2$ for $\sqrt{m} \leq \Delta \leq \frac{1}{4}m$.

4.3.2 The processes P_1 and P_2

We introduce a small modification to the definition of the processes P_1 and P_2. Namely, we leave the answering process for pair queries as described in Subsection 4.1.2 and modify the answering process for random new-neighbor queries as follows. Let $t \leq Q$, and π be a query-answer history of length $t-1$ such that π is consistent with G_1. If the t^{th} query is a new-neighbor query $q_t = u$ and $d^{kn}_{\pi}(u) < \frac{1}{2}\sqrt{m}$, then the processes P_1 and P_2 answer as described in Subsection 4.1.2. However, if the t^{th} query is a new-neighbor query $q_t = u$ such that $d^{kn}_{\pi}(u) \geq \frac{1}{2}\sqrt{m}$, then the processes answers as follows.

- The process P_1 answers with the set of all neighbors of u in G_1. That is, if u is in L, then the process replies with $a = R = \{r_1, \ldots, r_{\sqrt{m}}\}$, and if u is in R, then the process replies with $a = L = \{\ell_1, \ldots, \ell_{\sqrt{m}}\}$.

  The process P_2 answers with $a = \{v_1, \ldots, v_{\sqrt{m}}\}$, where $\{v_1, \ldots, v_{\sqrt{m}}\}$ is the set of neighbors of u in a subset of the graphs in $\mathcal{G}_2$. By the definition of $\mathcal{G}_2$, if u is in L, then this set is either R, or it is $R \setminus \{r_i\} \cup \{\ell_j\}$ for some $r_i \in R$ and $\ell_j \in L$, and if u is in R, then this set is either L, or it is $L \setminus \{\ell_i\} \cup \{r_j\}$ for some $\ell_i \in L$ and $r_j \in R$. For every such set $a \in Ans(\pi, q_t)$, the process returns a as an answer with probability

  $$\frac{|\mathcal{G}_2(\pi \circ (q_t, a))|}{|\mathcal{G}_2(\pi)|}.$$

  We call this query an all-neighbors query.

First note that the above modification makes the algorithm "more powerful". That is, every algorithm that is not allowed all-neighbors query can be emulated by an algorithm that is allowed this type of query. Therefore this only strengthen our lower bound results.

Also note that this modification does not affect the correctness of Lemma 4.1.1. We can redefine the function $\alpha_t(\pi)$ to be

$$\alpha_t(\pi) = \begin{cases} 1 & \text{if } q_t(\pi) \text{ is a pair query} \\ 1/\left(\sqrt{m} - d^{kn}_{\pi \le t-1}(u)\right) & \text{if } q_t(\pi) = u \text{ is a random new-neighbor query} \\ 1 & \text{if } q_t(\pi) \text{ is an all-neighbors query} \end{cases},$$

and the rest of the proof follows as before.

4.3.3 The auxiliary graph

For every $t \le Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and every pair (u,v), the witness graphs in $\mathcal{A}_{\pi,(u,v)}$ are graphs in which (u,v) is either a red pair or a blue pair. There is an edge between a witness graph W and a non-witness graph $\overline{W}$ if the two graphs have the same set of four-tuples except for two matched squares – one that contains the pair (u,v), $\langle u,v,u',v'\rangle$ and another one.

Definition 4.3.1 *We define a switch between a matched square and an unmatched square in the following manner. Let $\langle u,v,u',v'\rangle$ be a matched square and $\langle x,y,x',y'\rangle$ be an un matched squares. Informally, a switch between the squares is "unmatching" the matched square and instead "matching" the unmatched square.*

Formally, a switch consists of two steps. The first step is removing the edges (u,v) and (u',v') from the red matching M^C and the edges (u,u') and (v,v') from the blue matchings M^L and M^R respectively. The second step is adding the edges (x,y) and (x',y') from the red matching M^C and the edges (x,x') and (y,y') from the blue matchings M^L and M^R respectively. See Figure 8 for an illustration.

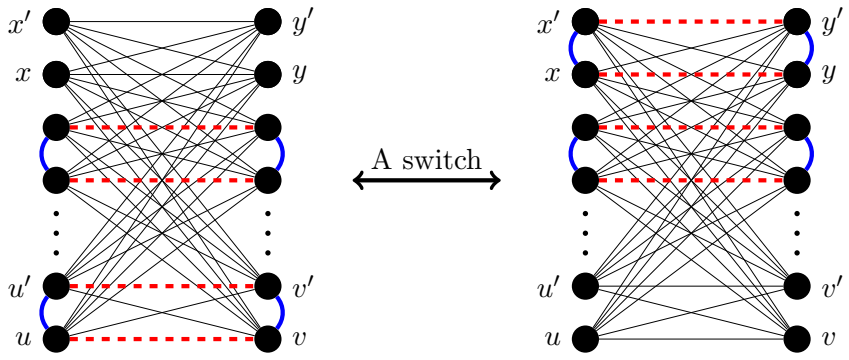


Figure 8: An illustration of a switch between the squares $\langle u,v,u',v'\rangle$ and $\langle x,y,x',y'\rangle$.

Lemma 4.3.1 *Let $\Delta = k \cdot \sqrt{m}$ for an integer k such that $1 < k \leq \frac{\sqrt{m}}{4}$ and let $Q = \frac{m^{3/2}}{600\Delta}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and every pair (u,v),*

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} = \frac{16k}{m} = \frac{16\Delta}{m^{3/2}}.$$

Proof: We start with proving that $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{2}m$. A witness graph in $\mathcal{A}_{\pi,(u,v)}$ with respect to a pair (u,v) is a graph in which (u,v) is part of a matched square $\langle u,v,u',v'\rangle$. Potentially, $\langle u,v,u',v'\rangle$ could be switched with every unmatched square to get a non-witness pair. There are $\sqrt{m}-k$ unmatched vertices on each side, so that there are $\binom{\sqrt{m}-k}{2} \cdot \binom{\sqrt{m}-k}{2} \geq \frac{1}{8}m^2$ potential squares. To get a graph that is in $\mathcal{G}_2(\pi)$, the unmatched square $\langle x,y,x',y'\rangle$ must be such that none of the induced pairs between the vertices x,x',y,y' have been observed yet by the algorithm. When all-neighbor queries are allowed, if at most Q queries has been performed, then at most $4Q$ pairs have been observed by the algorithm. Therefore, for at most $4\frac{m}{100k} \leq \frac{1}{4}m$ of the potential squares, an induced pair was queried. Hence, every witness square can be switched with at least $\frac{1}{8}m^2 - \frac{1}{4}m \geq \frac{1}{16}m^2$ consistent unmatched squares, implying that $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{16}m^2$.

To complete the proof it remains to show that $d_{nw}(\mathcal{A}_{\pi,(u,v)}) \leq mk$. To this end we would like to analyze the number of witness graphs that every non-witness $\overline{W}$ can be "turned" into. In every non-witness graph $\overline{W}$ the pair (u,v) is unmatched, and in order to turn $\overline{W}$ into a witness graph, one of the k matched squares should be removed and the pair (u,v) with an additional pair (u',v') should be "matched". There are k options to remove an existing square, and at most m options to choose a pair u',v' to match (u,v) with. Therefore, the number of potential neighbors of $\overline{W}$ is at most mk. It follows that

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} = \frac{16mk}{m^2} = \frac{16k}{m} = \frac{16\Delta}{m^{3/2}},$$

and the proof is complete. ■

4.3.4 Statistical distance

For an all-neighbors query $q = u$ we say that the corresponding answer is a witness answer if $u \in L$ and $a \neq R$, or symmetrically if $u \in R$ and $a \neq L$. Let E^Q be the set of all query-answer histories π of length Q such that there exists a query-answer pair (q,a) in π in which q is an all-neighbors pair and a is a witness answer with respect to that query, and let $\overline{E}^Q = \Pi^Q \setminus E^Q$. That is, $\overline{E}^Q$ is the set of all query-answer histories of length Q such that no all-neighbors query is answered with a witness answer. Let $\widetilde{P}_1$ and $\widetilde{P}_2$ by the induced distributions of the processes P_1 and P_2 conditioned on the event that the process do not reply with a witness answer. Observe that for every query-answer history π of length $t-1$, for every query q_t that is either a pair query or a random new-neighbor query and for every $a \in Ans(\pi, q_t)$,

$$\Pr_{\widetilde{P}_b}[a \mid \pi, q_t] = \Pr_{P_b}[a \mid \pi, q_t].$$

for $b \in \{1,2\}$. Therefore, the proof of the next lemma is exactly the same as the proof of Lemma 4.1.3, except that occurrences of the term $(\Delta/m^{3/2})$ are replaced by (k/m) instead of $(1/\sqrt{m})$ and we apply Lemma 4.3.1 instead of Lemma 4.1.2.

Lemma 4.3.2 *Let $\Delta = k \cdot \sqrt{m}$ for an integer k such that $1 < k \leq \frac{\sqrt{m}}{4}$ and let $Q = \frac{m^{3/2}}{600\Delta}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and for every pair or random new-neighbors query q_t,*

$$\sum_{a \in Ans(\pi, q_t)} \left| \Pr_{\widetilde{P}_1}[a \mid \pi, q_t] - \Pr_{\widetilde{P}_2}[a \mid \pi, q_t] \right| = \frac{96k}{m} = \frac{96\Delta}{m^{3/2}} \, .$$

Note that Lemma 4.3.2 does not cover all-neighbors queries, and hence we establish the next lemma.

Lemma 4.3.3 *Let $\Delta = k \cdot \sqrt{m}$ for an integer k such that $1 < k \leq \frac{\sqrt{m}}{4}$ and let $Q = \frac{m^{3/2}}{600\Delta}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and for every all-neighbors query q_t,*

$$\Pr_{P_2}[a_t \text{ is a witness answer } \mid \pi, q_t] \leq \frac{16k}{\sqrt{m}} \, .$$

Proof: Assume without loss of generality that $u \in L$. By the definition of the process P_2, it answers the query consistently with a uniformly selected random graph $G_2 \in \mathcal{G}_2(\pi)$ by returning the complete set of u's neighbors in G_2. In $\mathcal{G}_2$, there are two types of graphs. First, there are graphs in which u is not matched, that is $(u, u') \notin M^L$ for every vertex $u' \in L$. In these graphs the set of u's neighbors is $R = \{r_1, \ldots, r_{\sqrt{m}}\}$. We refer to these graphs as *non-witness graphs*. The second type of graphs are those in which $(u, u') \in M^L$ for some $u' \in L$ and $(u, v) \in M^C$ for some $v \in R$. In these graphs the set of u's neighbors is $(R \setminus \{v\}) \cup \{u'\}$. We refer to these graphs as *witness graphs*. As before, let $Ans(\pi, q_t)$ be the set of all possible answers for an all-neighbors query q_t. It holds that

$$\begin{aligned}
\Pr_{P_2}[a_t \text{ is a witness answer } \mid \pi, q_t] &= \sum_{\substack{a \in Ans(\pi, q_t) \\ a \neq R}} \Pr_{P_2}[a \mid \pi, q_t] \\
&= \sum_{u' \in L, v \in R} \frac{|\mathcal{G}_2(\pi \circ ((u,u'),1) \circ ((u,v),0))|}{|\mathcal{G}_2(\pi)|} \\
&= \sum_{u' \in L} \frac{|\mathcal{G}_2(\pi \circ ((u,u'),1))|}{|\mathcal{G}_2(\pi)|} \cdot \sum_{v \in R} \frac{|\mathcal{G}_2(\pi \circ ((u,u'),1) \circ ((u,v),0))|}{|\mathcal{G}_2(\pi)|} \\
&= \sum_{u' \in L} \frac{|\mathcal{G}_2(\pi \circ ((u,u'),1))|}{|\mathcal{G}_2(\pi)|} \, .
\end{aligned}$$

Similarly to the proof of Lemma 4.1.3, for every u and u' in L, $\frac{|\mathcal{G}_2(\pi \circ ((u,u'),1))|}{|\mathcal{G}_2(\pi)|} \leq \frac{16k}{m}$. Therefore,

$$\Pr_{P_2}[a_t \text{ is a witness answer } \mid \pi, q_t] = \sum_{u' \in L} \frac{|\mathcal{G}_2(\pi \circ ((u,u'),1))|}{|\mathcal{G}_2(\pi)|} \leq \sqrt{m} \cdot \frac{16k}{m} = \frac{16k}{\sqrt{m}} \, ,$$

and the lemma follows. ■

It remains to prove that a similar lemma to Lemma 4.1.4 holds for $\sqrt{m} \leq \Delta \leq \frac{1}{4}m$ (and the distributions $\mathcal{D}_1^{\text{ALG}}$ and $\mathcal{D}_2^{\text{ALG}}$ as defined in this subsection).

Lemma 4.3.4 *Let $\Delta = k \cdot \sqrt{m}$ for an integer k such that $1 < k \leq \frac{\sqrt{m}}{4}$. For every algorithm ALG that performs at most $Q = \frac{m^{3/2}}{600\Delta}$ queries, the statistical distance between $\mathcal{D}_1^{ALG}$ and $\mathcal{D}_2^{ALG}$ is at most $\frac{1}{3}$.*

Proof: Let the sets E^Q and $\overline{E}^Q$ be as defined in the beginning of this subsection. By the definition of the statistical distance, and since $\Pr_{P_1,\mathrm{ALG}}[E^Q] = 0$,

$$\begin{aligned} d(\mathcal{D}_1^{\mathrm{ALG}}, \mathcal{D}_2^{\mathrm{ALG}}) &= \frac{1}{2}\left(\sum_{\pi \in E^Q}\left|\Pr_{P_1,\mathrm{ALG}}[\pi] - \Pr_{P_2,\mathrm{ALG}}[\pi]\right| + \sum_{\pi \in \overline{E}^Q}\left|\Pr_{P_1,\mathrm{ALG}}[\pi] - \Pr_{P_2,\mathrm{ALG}}[\pi]\right|\right) \\ &= \frac{1}{2}\left(\Pr_{P_2,\mathrm{ALG}}[E^Q] + \sum_{\pi \in \overline{E}^Q}\left|\Pr_{P_1,\mathrm{ALG}}[\pi] - \Pr_{P_2,\mathrm{ALG}}[\pi]\right|\right). \end{aligned} \tag{91}$$

By Lemma 4.3.3, the probability of detecting a witness as a result of an all-neighbors query is at most $\frac{16k}{\sqrt{m}}$. Since in Q queries, there can be at most $4Q/\sqrt{m}$ all-neighbors queries, we have that

$$\Pr_{\mathcal{D}_2^{\mathrm{ALG}}}[\mathrm{E}^Q] \leq \frac{1}{6}\,. \tag{92}$$

We now turn to upper bound the second term. Let $\alpha = \Pr_{P_2,\mathrm{ALG}}[\mathrm{E}^Q]$.

$$\begin{aligned} \sum_{\pi \in \overline{E}^Q}\left|\Pr_{P_1,\mathrm{ALG}}[\pi] - \Pr_{P_2,\mathrm{ALG}}[\pi]\right| &= \sum_{\pi \in \overline{E}^Q}\left|\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi] \cdot \Pr_{P_1,\mathrm{ALG}}[\overline{E}^Q] - \Pr_{\widetilde{P}_2,\mathrm{ALG}}[\pi] \cdot \Pr_{P_2,\mathrm{ALG}}[\overline{E}^Q]\right| \\ &= \sum_{\pi \in \overline{E}^Q}\left|\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi] - (1-\alpha) \cdot \Pr_{\widetilde{P}_2,\mathrm{ALG}}[\pi]\right| \qquad (93) \\ &\leq \sum_{\pi \in \overline{E}^Q}\left|\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi] - \Pr_{\widetilde{P}_2,\mathrm{ALG}}[\pi]\right| + \alpha \cdot \Pr_{\widetilde{P}_2,\mathrm{ALG}}[\overline{E}^Q] \\ &\leq \sum_{\pi \in \overline{E}^Q}\left|\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi] - \Pr_{\widetilde{P}_2,\mathrm{ALG}}[\pi]\right| + \frac{1}{6}\,, \qquad (94) \end{aligned}$$

where in Equation (93) we used the fact that $\Pr_{P_1,\mathrm{ALG}}[\overline{E}^Q] = 1$, and in Equation (94) we used the fact that $\Pr_{\widetilde{P}_2,\mathrm{ALG}}[\overline{E}^Q] = 1$ and that $\alpha \leq 1/6$.

Therefore, it remains to bound

$$\sum_{\pi \in \overline{E}^Q}\left|\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi] - \Pr_{\widetilde{P}_2,\mathrm{ALG}}[\pi]\right|.$$

Let the hybrid distributions $D_{1,t}^{\mathrm{ALG}}$ for $t \in [Q-1]$ be as defined in Lemma 4.1.4 (based on the distributions $\mathcal{D}_1^{\mathrm{ALG}}$ and $\mathcal{D}_2^{\mathrm{ALG}}$ that are induced by the processes P_1 and P_2 that were defined in this subsection). Also, let $\widetilde{\mathcal{D}}_{1,t}^{\mathrm{ALG}}$ be the hybrid distribution $\mathcal{D}_{1,t}^{\mathrm{ALG}}$ conditioned on the event that no all-neighbors query is answered with a witness. That is, $\widetilde{\mathcal{D}}_{1,t}^{\mathrm{ALG}}$ is the distribution over query-answer histories π of length Q, where in the length t prefix ALG is answered by the process P_1, in the length $Q-t$ suffix ALG is answered by the process P_2, and each all-neighbors query is answered

consistently with G_1 (so that no witness is observed). By the above definitions and the triangle inequality,

$$\sum_{\pi\in\overline{E}^Q}\left|\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi]-\Pr_{\widetilde{P}_2,\mathrm{ALG}}[\pi]\right|\leq\sum_{t}^{Q-1}\sum_{\pi\in\overline{E}^Q}\left|\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t}}[\pi]\right|. \tag{95}$$

As in the proof of Lemma 4.1.4 we have that for every $t\in[Q-1]$,

$$\begin{aligned}&\sum_{\pi\in\overline{E}^Q}\left|\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t}}[\pi]\right|\\&\quad=\sum_{\substack{\pi'=\pi_1,\ldots,\pi_{t-1},q_t:\\ \pi'\in\overline{E}^{t-1}}}\Pr_{\widetilde{P}_1,\mathrm{ALG}}[\pi',q_t]\cdot\sum_{\substack{a\in Ans(\pi',q_t):\\ \pi'\circ(q_t,a)\in\overline{E}^t}}\left|\Pr_{\widetilde{P}_1}[a\,|\,\pi',q_t]-\Pr_{\widetilde{P}_2}[a\,|\,\pi',q_t]\right|.\end{aligned} \tag{96}$$

By Lemma 4.3.2 (and since for an all-neighbor query q_t we have that the (unique) answer according to $\widetilde{P}_2$ is the same as according to $\widetilde{P}_1$),

$$\sum_{\substack{a\in Ans(\pi',q_t):\\ \pi'\circ(q_t,a)\in\overline{E}^t}}\left|\Pr_{\widetilde{P}_1}[a\,|\,\pi',q_t]-\Pr_{\widetilde{P}_2}[a\,|\,\pi',q_t]\right|\leq\frac{96k}{m}=\frac{96\Delta}{m^{3/2}},$$

and it follows that

$$\sum_{\pi\in\overline{E}^Q}\left|\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t}}[\pi]\right|\leq\frac{96k}{m}=\frac{96\Delta}{m^{3/2}}.$$

Hence, for $Q=\frac{m^{3/2}}{600\Delta}$,

$$\sum_{t}^{Q-1}\sum_{\pi\in\overline{E}^Q}\left|\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t+1}}[\pi]-\Pr_{\widetilde{\mathcal{D}}^{\mathrm{ALG}}_{1,t}}[\pi]\right|\leq Q\cdot\frac{48\Delta}{m^{3/2}}\leq\frac{1}{6}. \tag{97}$$

Combining Equations (91), (92), (94), (95) and (97), we get

$$d(\mathcal{D}_1^{\mathrm{ALG}},\mathcal{D}_2^{\mathrm{ALG}})\leq\frac{1}{2}\left(\frac{1}{6}+\frac{1}{6}+\frac{1}{6}\right)\leq\frac{1}{3}, \tag{98}$$

and the proof is complete. ■

4.4 Lower Bound for $\Delta<\frac{1}{4}\sqrt{m}$.

4.4.1 The construction

In this case the basic structure of G_1 and $\mathcal{G}_2$ is a bit different. Also, for the sake of simplicity, we present graphs with $2m$ edges, and either 0 or 4Δ triangles. The graph G_1 has three components – two complete bipartite graphs, each over $2\sqrt{m}$ vertices, and an independent set of size $n-4\sqrt{m}$. Let A and B be the left-hand side and the right-hand side sets, respectively, of the first bipartite component, and C and D of the second one. We refer to the edges between A and B and the edges

between C and D as black edges. We divide each of these sets into $\frac{\sqrt{m}}{\Delta}$ subsets of size Δ, denoted $\{\Lambda_1, \ldots, \Lambda_{\frac{\sqrt{m}}{\Delta}}\}$ for $\Lambda \in \{A, B, C, D\}$. For every $1 \leq i \leq \frac{\sqrt{m}}{\Delta}$, we first remove a complete bipartite graph between A_i and B_i and between C_i and D_i, and refer to the removed edges as red edges. We then add a complete bipartite graph between B_i and C_i and between D_i and A_i, and refer to added edges as blue edges. Note that this maintains the degrees of all the vertices to be $\sqrt{m}$.

In $\mathcal{G}_2$ the basic structure of all the graphs is the same as of G_1 with the following modifications. Each graph is defined by the choice of four "special" vertices a^*, b^*, c^*, d^* such that $a^* \in A_{i_{a^*}}, b^* \in B_{i_{b^*}}, c^* \in C_{i_{c^*}}$ and $d^* \in D_{i_{d^*}}$ for some indices $i_{a^*}, i_{b^*}, i_{c^*}$ and i_{d^*} such that no two indices are equal. We then add edges (a^*, c^*) and (b^*, d^*), referred to as green edges, and remove edges (a^*, b^*) and (c^*, d^*), referred to as purple edges. We also refer to the green and purple edges as special edges. Note that we add one edge and remove one edge from each special vertex, thus maintaining their initial degrees. See Figure 9.

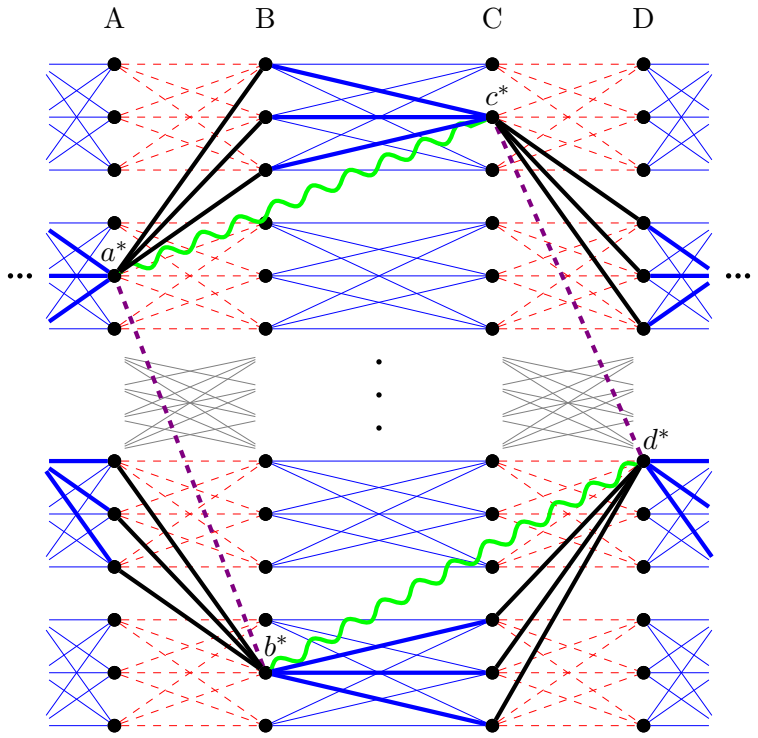


Figure 9: An illustration of a graph in $\mathcal{G}_2$. The broken thin (red) edges describe edges that were removed and the thin (blue) edges describe edges that were added. The broken thick (purple) edges describe the special non-edges (a^*, b^*) and (c^*, d^*). The curly (green) edges describe the special edges (a^*, c^*) and (b^*, d^*).

We first prove that $\Delta(G_1) = 0$ and then that for every graph G in $\mathcal{G}_2$, $\Delta(G) = 4\Delta$.

Claim 4.4.1 *The graph G_1 has no triangles.*

Proof: Consider an edge (u, v) in G_1. First assume u and v are connected by a black edge, that is, they are on different sides of the same bipartite component. Hence we can assume without loss

of generality that $u \in A$ and that $v \in B$. Since u is in A it is only connected to vertices in B or vertices in D. Since v is in B it is only connected to vertices in A or vertices in C. Thus u and v cannot have a common neighbor. A similar analysis can be done for a pair (u, v) that is connected by a blue edge. Therefore $\Delta(G)$ is indeed zero as claimed. ■

Claim 4.4.2 *For every graph $G \in \mathcal{G}_2$, $\Delta(G) = 4\Delta$.*

Proof: Since the only differences between G_1 and graphs in $\mathcal{G}_2$ are the two added green edges and the two removed red edges, any triangle in $\mathcal{G}_2$ must include a green edge. Therefore we can count all the triangles that the green edges form. Consider the green edge (a^*, c^*) and recall that a^* is in $A_{i_{a^*}}$ and c^* is in $C_{i_{c^*}}$. The only common neighbors of (a^*, c^*) are all the vertices in $B_{i_{c^*}}$ and all the vertices in $D_{i_{a^*}}$. A vertex v such that $v \notin B_{i_{c^*}}$ and $v \notin D_{i_{a^*}}$ is either (1) in A or in $D \setminus D_{i_{a^*}}$, in which case it is not a neighbor of a^*, or it is (2) in C or in $B \setminus B_{i_{c^*}}$, in which case it is not a neighbor of c^*. Since both $B_{i_{c^*}}$ and $D_{i_{a^*}}$ are of size Δ, the edge (a^*, c^*) participates in 2Δ triangles. Similarly the edge (b^*, d^*) participate in 2Δ triangles, and together we get that $\Delta(G) = 4\Delta$, as claimed. ■

4.4.2 The processes P_1 and P_2

The definition of the processes P_1 and P_2 is the same as in Subsection 4.3.2 (using the modified definitions of G_1 and $\mathcal{G}_2$).

4.4.3 The auxiliary graph

We define a switch for this case as well. Informally, a switch between a matched pair (u^*, v^*) and an unmatched pair (u, v) is "unmatching" (u^*, v^*) and "matching" (u, v) instead. Formally stating we define a switch as follows.

Definition 4.4.1 *A switch between a green pair (a^*, c^*) and a pair (a, c) such that $a \in A_i$, $c \in C_j$ and none of the indices i, j, i_{b^*}, i_{d^*} are equal, is the following two steps process. In the first step we "unmatch" (a^*, c^*) by removing the green edge (a^*, c^*) and adding the edges (a^*, b^*) and (c^*, d^*). In the second step we "match" (a, c) by adding the green edge (a, c) and removing the edges (a, b^*) and (c, d^*). A switch with the pair (b^*, d^*) can be defined in a similar manner.*

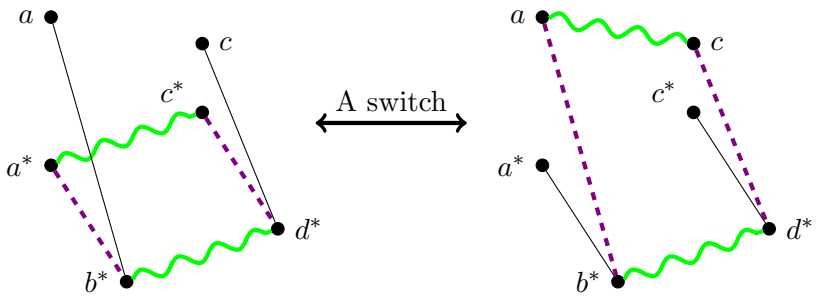


Figure 10: An illustration of a switch between the pairs (a^*, c^*) and (a, c).

Let $\Delta < \sqrt{m}$ and let $Q = \frac{m}{600}$. For every $t \leq Q$, every query-answer history π of length $t-1$ and every pair (u, v) we define the following auxiliary graph. The witness nodes are graphs in which (u, v) is one of the four special pairs. If the pair is a green matched pair then there is an edge in the auxiliary graph between a witness graph W and a non-witness graph $\overline{W}$, if $\overline{W}$ can be obtained from W by a single switch between (u, v) and another unmatched pair.

Lemma 4.4.3 *For $\Delta < \frac{1}{4}\sqrt{m}$ let $Q = \frac{m}{600}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and every pair (u, v),*

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} = \frac{8}{m}.$$

Proof: We analyze the case where the pair (u, v) is such that $u \in A$ and $v \in C$, as the proof for the other cases is almost identical. We first prove that $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{8}m$. A witness graph W is a graph in which (a, c) is a special pair. That is $(u, v) = (a^*, c^*)$. Potentially, for every pair (a', c') such that $a' \in A_i$, $c' \in C_j$ and none of the indices i, j, i_{b^*}, i_{d^*} are equal, the graph resulting from a switch between (a^*, c^*) and (a', c') is a non-witness graph. There are $\sqrt{m} - 2\Delta$ vertices a' in $A \setminus (A_{i_{b^*}} \cup A_{i_{d^*}})$ and for each such a' there are $\sqrt{m} - 3\Delta$ vertices c' in $C \setminus (C_{i_{b^*}} \cup C_{i_{d^*}} \cup C_{i_{a'}})$. Since $\Delta < \frac{1}{4}\sqrt{m}$, there are at least $(\sqrt{m} - 2\Delta) \cdot (\sqrt{m} - 3\Delta) = m - 6\Delta^2 \geq \frac{1}{4}m$ potential pairs (a', c') that (a^*, c^*) could be switched with. For the resulting graph to be consistent, that is, to be in $\mathcal{G}_2(\pi)$, the pair (a', c') must be such that the pairs (a', c'), (a^*, b^*) and (c^*, d^*) have not been observed yet by the algorithm. Since the number of queries is at most $\frac{1}{600}m$, at least $\frac{1}{4}m - \frac{1}{125}m \geq \frac{1}{8}m$ of the potential pairs (a', c') can be switched with (a^*, c^*) such that the resulting graph is consistent with $\mathcal{G}_2(\pi)$. Therefore, $d_w(\mathcal{A}_{\pi,(u,v)}) \geq \frac{1}{8}m$.

Now consider a non-witness graph $\overline{W}$. There is only one possibility to turn $\overline{W}$ into a witness graph, which is to switch the pair (u, v) with the green pair (a^*, c^*). Therefore, the maximal degree of every non-witness graph, $d_{nw}(\mathcal{A}_{\pi,(u,v)})$, is 1.

Together we get that

$$\frac{d_{nw}(\mathcal{A}_{\pi,(u,v)})}{d_w(\mathcal{A}_{\pi,(u,v)})} \leq \frac{8}{m},$$

and the proof is complete. ■

4.4.4 Statistical distance

A similar proof to the ones of Lemma 4.3.2 and Lemma 4.3.3 using Lemma 4.4.3 gives the following lemmas for the case that $1 \leq \Delta < \frac{1}{4}\sqrt{m}$.

Lemma 4.4.4 *Let $1 \leq \Delta < \frac{1}{4}\sqrt{m}$ and $Q = \frac{m}{600}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and for every all-neighbors query q_t,*

$$\Pr_{P_2}[a_t \text{ is a witness answer } | \pi, q_t] \leq \frac{16}{m}.$$

Lemma 4.4.5 *Let $1 \leq \Delta < \frac{1}{4}\sqrt{m}$ and $Q = \frac{m}{600}$. For every $t \leq Q$, every query-answer history π of length $t-1$ such that π is consistent with G_1 and for every pair or random new-neighbors query q_t,*

$$\sum_{a \in Ans(\pi, q_t)} \left| \Pr_{\widetilde{P}_1}[a \,|\, \pi, q_t] - \Pr_{\widetilde{P}_2}[a \,|\, \pi, q_t] \right| = \frac{96}{m}.$$

The next lemma is proven in a similar way to 1.3.4 based on the above two lemma.

Lemma 4.4.6 *Let $1 \leq \Delta < \frac{1}{4}\sqrt{m}$. For every algorithm ALG that asks at most $Q = \frac{m}{600}$, the statistical distance between $\mathcal{D}_1^{ALG}$ and $\mathcal{D}_2^{ALG}$ is at most $\frac{1}{3}$.*

4.5 Wrapping things up

Theorem 4.2 follows from Lemmas 4.1.4, 4.2.3, 4.3.4 and 4.4.6, and the next corollary is proved using Theorems 4.2 and 4.1.

Corollary 4.3 *Any multiplicative-approximation algorithm for the number of triangles in a graph must perform $\Omega\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right)$ queries, where the allowed queries are degree queries, pair queries and neighbor queries.*

Proof: Assume towards a contradiction that there exists an algorithm ALG' for which the following holds:

1. ALG' is allowed to ask neighbor queries as well as degree queries and pair queries.
2. ALG' asks Q' queries.
3. ALG' outputs a $(1 \pm \epsilon)$-approximation to the number of triangles of any graph G with probability greater than $2/3$.

Using ALG' we can define an algorithm ALG that is allowed random new-neighbor queries, performs at most $Q = 3Q'$ queries and answers correctly with the same probability as ALG' does. ALG runs ALG' and whenever ALG' performs a query q'_t, ALG does as follows:

- If q'_t is a degree query, ALG performs the same query and sets $a'_t = a_t$.
- If q'_t is a pair query (u, v), then ALG performs the same query $q = q'$. Let a_t be the corresponding answer.
  - If $a_t = 0$, then ALG sets $a'_t = a_t$.
  - If $a_t = 1$, then ALG sets $a'_t = (a_t, i, j)$, such that i and j are randomly chosen labels that have not been previously used for neighbors of u and v, and are within the ranges $[1..d(u)]$ and $[1..d(v)]$ respectively.
- If q'_t is a neighbor query (u, i), ALG performs a random new-neighbor query $q_t = u$, and returns the same answer $a'_t = a_t$.

We note that the above requires the algorithm ALG to store for every vertex v, all the labels used for its neighbors in the previous steps. Once ALG' outputs an answer, ALG outputs the same answer. It follows that ALG performs at most $3Q$ queries to the graph G. By the third assumption above, ALG outputs a $(1 \pm \epsilon)$-approximation to the number of triangles of any graph G with probability greater than $2/3$. If $Q' \notin \Omega\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right)$ then $Q \notin \Omega\left(\frac{n}{\Delta(G)^{1/3}} + \min\left\{m, \frac{m^{3/2}}{\Delta(G)}\right\}\right)$ which is a contradiction to Theorem 4.1 and Theorem 4.2. Therefore, the corollary follows. ■

References

[Avr10] Haim Avron. Counting triangles in large graphs using randomized matrix trace estimation. In *Workshop on Large-scale Data Mining: Theory and Applications*, 2010. 1

[AYZ97] Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. 1

[BBCG08] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–24. ACM, 2008. 4

[BFL+06] Luciana S Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. Counting triangles in data streams. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 253–262. ACM, 2006. 4

[BYKS02] Ziv Bar-Yossef, Ravi Kumar, and D Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 623–632. Society for Industrial and Applied Mathematics, 2002. 4

[CC11] Shumo Chu and James Cheng. Triangle listing in massive networks and its applications. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 672–680. ACM, 2011. 1

[CEF+05] Artur Czumaj, Funda Ergün, Lance Fortnow, Avner Magen, Ilan Newman, Ronitt Rubinfeld, and Christian Sohler. Approximating the weight of the euclidean minimum spanning tree in sublinear time. *SIAM Journal on Computing*, 35(1):91–109, 2005. 4

[CN85] Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on Computing*, 14(1):210–223, 1985. 1

[Col88] James S Coleman. Social capital in the creation of human capital. *American journal of sociology*, pages S95–S120, 1988. 1

[CRT05] Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM Journal on Computing*, 34(6):1370–1379, 2005. 4

[CS09] Artur Czumaj and Christian Sohler. Estimating the weight of metric minimum spanning trees in sublinear time. *SIAM Journal on Computing*, 39(3):904–922, 2009. 4

[EM02] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the national academy of sciences*, 99(9):5825–5829, 2002. 1

[Fei06] Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984, 2006. 3

[FWVDC10] Brooke Foucault Welles, Anne Van Devender, and Noshir Contractor. Is a friend a friend?: Investigating the structure of friendship networks in virtual worlds. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 4027–4032. ACM, 2010. 1

[GR02] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, pages 302–343, 2002. 59

[GR08] Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Structures and Algorithms*, 32(4):473–493, 2008. 2, 3, 4

[GRS11] Mira Gonen, Dana Ron, and Yuval Shavitt. Counting stars and other small subgraphs in sublinear-time. *SIAM Journal on Discrete Math*, 25(3):1365–1411, 2011. 1, 2, 3, 4

[HKNO09] Avinatan Hassidim, Jonathan A Kelner, Huy N Nguyen, and Krzysztof Onak. Local graph partitions for approximation and testing. In *Proceedings of the Fiftieth Annual Symposium on Foundations of Computer Science (FOCS)*, pages 22–31. IEEE, 2009. 4

[IR78] Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978. 1

[JG05] Hossein Jowhari and Mohammad Ghodsi. New streaming algorithms for counting triangles in graphs. In *Computing and Combinatorics*, pages 710–716. Springer, 2005. 4

[KMPT12] Mihail N Kolountzakis, Gary L Miller, Richard Peng, and Charalampos E Tsourakakis. Efficient triangle counting in large graphs via degree-based vertex partitioning. *Internet Mathematics*, 8(1-2):161–185, 2012. 4

[Lat08] Matthieu Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science*, 407(1):458–473, 2008. 1

[MR09] Sharon Marko and Dana Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Transactions on Algorithms*, 5(2), 2009. 4

[MSOI+02] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002. 1

[NO08] Huy N Nguyen and Krzysztof Onak. Constant-time approximation algorithms via local improvements. In *Proceedings of the Forty-Ninth Annual Symposium on Foundations of Computer Science (FOCS)*, pages 327–336. IEEE, 2008. 4

[ORRR12] Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1123–1131. SIAM, 2012. 4

[Por00] Alejandro Portes. Social capital: Its origins and applications in modern sociology. *LESSER, Eric L. Knowledge and Social Capital. Boston: Butterworth-Heinemann*, pages 43–67, 2000. 1

[PR07] Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1-3):183–196, 2007. 4

[Sch07] Thomas Schank. Algorithmic aspects of triangle-based network analysis. *Phd in computer science, University Karlsruhe*, 2007. 1

[SPK14] C Seshadhri, Ali Pinar, and Tamara G Kolda. Wedge sampling for computing clustering coefficients and triangle counts on large graphs. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):294–307, 2014. 1

[TDM+09] Charalampos E Tsourakakis, Petros Drineas, Eirinaios Michelakis, Ioannis Koutis, and Christos Faloutsos. Spectral counting of triangles in power-law networks via element-wise sparsification. In *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, pages 66–71. IEEE, 2009. 4

[TKM11] Charalampos E Tsourakakis, Mihail N Kolountzakis, and Gary L Miller. Triangle sparsifiers. *J. Graph Algorithms Appl.*, 15(6):703–726, 2011. 4

[TKMF09] Charalampos E Tsourakakis, U Kang, Gary L Miller, and Christos Faloutsos. Doulion: counting triangles in massive graphs with a coin. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 837–846. ACM, 2009. 4

[Tso08] Charalampos E. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pages 608–617. IEEE, 2008.

[Was94] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994. 1

[YK11] Jin-Hyun Yoon and Sung-Ryul Kim. Improved sampling for triangle counting with mapreduce. In *Convergence and Hybrid Information Technology*, pages 685–689. Springer, 2011. 4

[YYI09] Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. An improved constant-time approximation algorithm for maximum. In *Proceedings of the Fourty-First Annual ACM Symposium on the Theory of Computing*, pages 225–234. ACM, 2009. 4

ECCC ISSN 1433-8092

http://eccc.hpi-web.de