SIAM J. Discrete Math.
Vol. 25, No. 3, pp. 1365–1411


# COUNTING STARS AND OTHER SMALL SUBGRAPHS IN SUBLINEAR-TIME*

MIRA GONEN†, DANA RON‡, AND YUVAL SHAVITT§

**Abstract.** Detecting and counting the number of copies of certain subgraphs (also known as *network motifs* or *graphlets*) is motivated by applications in a variety of areas ranging from biology to the study of the World Wide Web. Several polynomial-time algorithms have been suggested for counting or detecting the number of occurrences of certain network motifs. However, a need for more efficient algorithms arises when the input graph is very large, as is indeed the case in many applications of motif counting. In this paper we design *sublinear-time* algorithms for approximating the number of copies of certain constant-size subgraphs in a graph $G$. That is, our algorithms do not read the whole graph, but rather query parts of the graph. Specifically, we consider algorithms that may query the degree of any vertex of their choice and may ask for any neighbor of any vertex of their choice. The main focus of this work is on the basic problem of counting the number of length-2 paths and more generally on counting the number of stars of a certain size. Specifically, we design 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 stars of size $s+1$ in the graph. The expected query complexity and running time of the algorithm are $O\Big(\frac{n}{(\nu_s(G))^{\frac{1}{s+1}}} + \min\Big\{n^{1-\frac{1}{s}}, \frac{n^{s-\frac{1}{s}}}{(\nu_s(G))^{1-\frac{1}{s}}}\Big\}\Big) \cdot \mathrm{poly}(\log n, 1/\epsilon)$. We also prove lower bounds showing that this algorithm is tight up to polylogarithmic factors in $n$ and the dependence on $\epsilon$. Our work extends the work of Feige [*SIAM J. Comput.*, 35 (2006), pp. 964–984] and Goldreich and Ron [*Random Structures Algorithms*, 32 (2008), pp. 473–493] on approximating the number of edges (or average degree) in a graph. Combined with these results, our result can be used to obtain an estimate on the variance of the degrees in the graph and corresponding higher moments. In addition, we give some (negative) results on approximating the number of triangles and on approximating the number of length-3 paths in sublinear-time.

**Key words.** sublinear-time algorithms, approximate counting, subgraphs

**AMS subject classifications.** 68Q17, 68Q25, 68R10

**DOI.** 10.1137/100783066

## 1. Introduction.

This work is concerned with approximating the number of copies of certain constant-size subgraphs in a graph $G$. Detecting and counting subgraphs (also known as *network motifs* [MSOI+02] or *graphlets* [PCJ04]) is motivated by applications in a variety of areas ranging from biology to the study of the World Wide Web (see, e.g., [MSOI+02], [KIMA04], [SIKS06], [PCJ04], [Wer06], [SSRS06], [GK07], [DSG+08], [HBPS07], [ADH+08], [HA08], [GS09]), as well as by the basic quest to understand simple structural properties of graphs. Our work differs from previous works on counting subgraphs (with the exception of counting the number of edges [Fei06], [GR08]) in that we design *sublinear* algorithms. That is, our algorithms do not read the whole graph, but rather query parts of the graph (where we shall specify the type of queries we allow when we state our precise results). The need for such algorithms arises when the input graph is very large (as is indeed the case in many of the application of motif counting).

*Received by the editors January 19, 2010; accepted for publication (in revised form) June 13, 2011; published electronically September 20, 2011.

http://www.siam.org/journals/sidma/25-3/78306.html

†Department of Mathematics, Bar Ilan University (gonenm1@math.biu.ac.il). This work was done while the author was at Tel Aviv University.

‡School of Electrical Engineering, Tel-Aviv University (danar@eng.tau.ac.il). This research was supported by the Israel Science Foundation (grant no. 246/08).

§School of Electrical Engineering, Tel-Aviv University (shavitt@eng.tau.ac.il). This research was supported by the Israel Science Foundation Center of Excellence Program (grant 1685/07).

The main focus of this work is on the problem of counting the number of length-2 paths and more generally on counting the number of stars of a certain size. We emphasize that we count *noninduced* subgraphs. We shall use the term *s-star* for a subgraph over $s+1$ vertices in which one single vertex (the star *center*) is adjacent to all other vertices (and there are no edges between the other vertices). Observe that a length-2 path is a 2-star. We also give some (negative) results on approximating the number of triangles and on approximating the number of length-3 paths.

As we show in detail below, we obtain almost matching upper and lower bounds on the query complexity and running time of approximating the number of $s$-stars. These bounds are a function of the number, $n$, of graph vertices and the actual number of $s$-stars in the graph, and have a nontrivial form. Our results extend the works [Fei06] and [GR08] on sublinear-time approximation of the average degree in a graph, or equivalently, approximating the number of edges (where an edge is the simplest (nonempty) subgraph). Note that if we have an estimate for the number of length-2 paths and for the average degree, then we can obtain an estimate for the variance of the degrees in the graph, and the number of larger stars corresponds to higher moments. Thus, the study of the frequencies of these particular subgraphs in a graph sheds light on basic structural properties of graphs.

**Our results.** We assume graphs are represented by the incidence lists of the vertices (or, more precisely, incidence arrays), where each list is accompanied by its length. Thus, the algorithm can query the degree, $d(v)$, of any vertex $v$ of its choice (a *degree query*), and for any vertex $v$ and index $1 \le i \le d(v)$, it can query who is the $i$th neighbor of $v$ (a *neighbor query*).

Let $\nu_s(G)$ denote the number of $s$-stars in a graph $G$. Our main positive result is 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 (over the coin flips of the algorithm), $(1-\epsilon)\nu_s(G) \le \hat{\nu}_s \le (1+\epsilon)\nu_s(G)$. The expected query complexity and running time of the algorithm are

$$O\left(\frac{n}{(\nu_s(G))^{\frac{1}{s+1}}} + \min\left\{n^{1-\frac{1}{s}}, \frac{n^{s-\frac{1}{s}}}{(\nu_s(G))^{1-\frac{1}{s}}}\right\}\right) \cdot \mathrm{poly}(\log\, n, 1/\epsilon). \tag{1.1}$$

The dependence on $s$ is exponential, and is not stated explicitly as we assume $s$ is a constant.

The complexity of our algorithm as stated in (1.1) is best understood by viewing Table 1.1, in which we see that there are three regions when considering $\nu_s(G)$ as a function of $n$, and in each the complexity is governed by a different term. Observe the following:

- In the first range ($\nu_s(G) \le n^{1+\frac{1}{s}}$), the complexity of the algorithm (which is at its maximum when $\nu_s(G)$ is very small) decreases as $\nu_s(G)$ increases.

TABLE 1.1
*The query complexity and running time of our algorithm for approximating the number of s-stars.*

| $\nu_s(G)$ | Query and time complexity |
|---|---|
| $\nu_s(G) \le n^{1+\frac{1}{s}}$ | $O\left(\frac{n}{(\nu_s(G))^{\frac{1}{s+1}}}\right) \cdot \mathrm{poly}(\log\, n, 1/\epsilon)$ |
| $n^{1+\frac{1}{s}} < \nu_s(G) \le n^s$ | $O\left(n^{1-\frac{1}{s}}\right) \cdot \mathrm{poly}(\log\, n, 1/\epsilon)$ |
| $\nu_s(G) > n^s$ | $O\left(\frac{n^{s-\frac{1}{s}}}{(\nu_s(G))^{1-\frac{1}{s}}}\right) \cdot \mathrm{poly}(\log\, n, 1/\epsilon)$ |

- In the second range ($n^{1+\frac{1}{s}} < \nu_s(G) \leq n^s$), the complexity does not depend on $\nu_s(G)$.
- In the last range ($\nu_s(G) > n^s$), it again decreases as $\nu_s(G)$ increases (where in the extreme case, when $\nu_s(G) = \Omega(n^{s+1})$, the complexity is just poly(log $n, 1/\epsilon$)).

For example, for $s = 3$ and a constant $\epsilon$, if $\nu_3(G) = \Theta(n)$, then the query complexity and running time of the algorithm are $\tilde{O}(n^{3/4})$, if $\nu_3(G) = \Theta(n^2)$, then the query complexity and running time are $\tilde{O}(n^{2/3})$, and if $\nu_3(G) = \Theta(n^4)$, then the query complexity and running time of the algorithm are poly(log $n$).

The expression in (1.1) might seem unnatural and hence merely an artifact of our algorithm. However, we prove that it is tight up to polylogarithmic factors in $n$ and the dependence on $\epsilon$. Namely, we show the following:

- Any multiplicative approximation algorithm for the number of $s$-stars must perform $\Omega\left(\frac{n}{(\nu_s(G))^{\frac{1}{s+1}}}\right)$ queries.
- Any constant-factor approximation algorithm for the number of $s$-stars must perform $\Omega(n^{1-\frac{1}{s}})$ queries when the number of $s$-stars is $O(n^s)$.
- Any constant-factor approximation algorithm for the number of $s$-stars must perform $\Omega\left(\frac{n^{s-\frac{1}{s}}}{(\nu_s(G))^{1-\frac{1}{s}}}\right)$ queries when the number of $s$-stars is $\Omega(n^s)$.

We mention that another type of queries, which are natural in the context of dense graphs, are vertex-pair queries. That is, the algorithm may query about the existence of an edge between any pair of vertices. We note that our lower bounds imply that allowing such queries cannot reduce the complexity for counting the number of stars (except possibly by polylogarithmic factors in $n$).

Finally, we consider other small graphs that extend length-2 paths: triangles and length-3 paths. We show that if an algorithm uses a number of queries that is sublinear in the number of edges, then for triangles it is hard to distinguish between the case that a graph contains $\Theta(n)$ triangles and the case that it contains *no* triangles, and for length-3 paths it is hard to distinguish between the case that there are $\Theta(n^2)$ length-3 paths and the case that there are no such paths. These lower bounds hold when the number of edges is $\Theta(n)$.

**Techniques.** Our starting point is similar to the one in [GR08]. Consider a partition of the graph vertices into $O(\log n/\epsilon)$ *buckets*, where in each bucket all vertices have the same degree (with respect to the entire graph) up to a multiplicative factor of $(1 \pm O(\epsilon))$. (For a precise definition of the buckets, see section 3.1.) If we could get a good estimate of the size of each bucket by sampling, then we would have a good estimate of the number of $s$-stars (since the vertices in each bucket are the centers of approximately the same number of stars). The difficulty is that some buckets may be very small and we might not even hit them when sampling vertices. The approach taken in [GR08] to get a multiplicative estimate of $(1 \pm \epsilon)$ is to estimate the number of edges between large buckets and small buckets, and incorporate this estimate into the final approximation.[1]

[1]We note that in the case of the average degree (number of edges), if we ignore the small buckets (for an appropriate definition of "small"), then we can already get (roughly) a factor-2 approximation in $O(\sqrt{n})$ time [Fei06], [GR08]. However, this is not the case for $s$-stars (even when $s = 2$). To verify this, consider the case that the graph $G$ is a star. There are two buckets: one containing only the star center, and another containing all other vertices. If we ignore the (very) small bucket that contains the star center, then we get an estimate of 0 while the graph contains $\Theta(n^2)$ length-2 paths (2-stars).

Here we first observe that we need a more refined procedure. In particular, we need a separate estimate for the number of edges between each large bucket and each small bucket. Note that if we have an estimate $\hat{e}$ of the number of edges incident to vertices in a certain bucket, and all vertices in that bucket have degree roughly $d$, then the number of $s$-stars whose center belongs to this bucket is approximately $\frac{1}{s}\hat{e} \cdot \binom{d-1}{s-1}$. To see why this is true, consider an edge $(u, v)$ that is incident to a vertex $u$ that has degree (roughly) $d$. Then the number of stars that include this edge and are centered at $u$ is (roughly) $\binom{d-1}{s-1}$. If we sum this expression over all $\hat{e}$ edges that are incident to vertices in the bucket of $u$, then each star (that is centered at a vertex in the bucket) is counted $s$ times, and hence we divide the expression $\hat{e} \cdot \binom{d-1}{s-1}$ by $s$.

As a first attempt for obtaining such an estimate on the number of edges incident to vertices in a bucket, consider uniformly sampling edges incident to vertices that belong to large buckets. We can then estimate the number of edges between the large buckets and each small bucket by querying the degree of the other endpoint of each sampled edge. It is possible to show that for a sufficiently large sample of edges we can indeed obtain a good estimate for the number of $s$-stars using this procedure. However, the complexity of the resulting procedure, which is dominated by the number of edges that need to be sampled, is far from optimal. The reason for this has to do with the variance between the number of edges that different vertices in the same large bucket have to the various small buckets. To overcome this and get an (almost) optimal algorithm, we further refine the sampling process.

Specifically, we first define the notion of *significant* small buckets. Such buckets have a nonnegligible contribution to the total number of $s$-stars (where each vertex accounts for the number of stars that it is a center of). Now, for each large bucket $B_i$ and (significant) small bucket $B_j$, we further consider partitioning the vertices in $B_i$ according to the number of neighbors they have in $B_j$. The difficulty is that in order to determine *exactly* to which subbucket a vertex in $B_i$ belongs, we would need to query all its neighbors, which may be too costly. Moreover, even if an estimate on this number suffices, if a vertex in $B_i$ has relatively few neighbors in $B_j$, then we would need a relatively large sample of its neighbors in order to obtain such an estimate. Fortunately, we encounter a tradeoff between the number of vertices in $B_i$ that need to be sampled in order to get sufficiently many vertices that belong to a particular subbucket and the number of neighbors that should be sampled so as to detect (approximately) to which subbucket a vertex belongs. We exemplify this by an extreme case: consider the subbucket of vertices for which at least half of their neighbors belong to $B_j$. This subbucket may be relatively small (and still contribute significantly to the total number of edges between $B_i$ and $B_j$), but if we sample a vertex from this subbucket, then we can easily detect this by taking only a constant sample of its neighbors. For more details, see subsection 3.4.

**Related work.** As noted previously, our work extends the works [Fei06], [GR08] on approximating the average degree of a graph in sublinear-time. In particular, our work is most closely related to [GR08], where it is shown how to get an estimate of the average degree of a graph $G$ that is within $(1 \pm \epsilon)$ of the correct value $\bar{d}(G)$. The expected running time and query complexity of the algorithm in [GR08] are $O((n/\bar{d}(G))^{1/2}) \cdot \text{poly}(\log n, 1/\epsilon)$.

There are quite a few works that deal with finding subgraphs of a certain kind and of counting their number in polynomial-time. One of the most elegant techniques devised is *color-coding*, introduced in [AYZ95], and further applied in [AYZ97], [AR02], [AG10], [ADH+08], [AG09]. In particular, in [AR02] the authors use color-coding and a technique from [KL83] to design a randomized algorithm for approximately counting the number

of subgraphs in a given graph $G$ that are isomorphic to a bounded treewidth graph $H$. The running time of the algorithm is $k^{O(k)} \cdot n^{b+O(1)}$, where $n$ and $k$ are the number of vertices in $G$ and $H$, respectively, and $b$ is the treewidth of $H$. In [AG10] the authors use color-coding and balanced families of perfect hash functions to obtain a deterministic algorithm for approximately counting simple paths or cycles of size $k$ in time $2^{O(k \log \log k)} n^{O(1)}$. In [ADH+08]these results are improved in terms of the dependence on $k$. We note that sampling is also applied in [KIMA04], [Wer06], where the authors are interested in uniformly sampling induced subgraphs of a given size $k$. Other related work in this category includes [DLR95], [GK07], [BBCG08], [Kou08], [Wil09], [GS09], [BHKK09], [AFS09], [KW09], [VW09]. In [FG04] the authors conclude that most likely there is no $f(k) \cdot n^c$ algorithm for *exactly* counting cycles or paths of length $k$ in a graph of size $n$ for any computable function $f: N \to N$ and constant $c$.

Another related line of work deals with approximating other graph measures (such as the weight of a minimum spanning tree) in sublinear-time and includes [CRT05], [CS09], [CEF+05], [PR07], [NO08], [YYI09].

**Organization.** For the sake of the exposition, we first describe the algorithm and the analysis, as well as the lower bounds, for the case $s = 2$, that is, length-2 paths. This is done in sections 3 and 4, respectively. In section 5 we explain how to adapt the algorithm for length-2 paths in order to get an algorithm for $s$-stars, and in section 6 we explain how to adapt the lower bounds. Finally, in section 7 we shortly discuss triangles and length-3 paths.

**2. Preliminaries.** Let $G = (V, E)$ be an undirected graph with $|V| = n$ vertices and $|E| = m$ edges, where $G$ is simple so that it contains no multiple edges. We denote the set of neighbors of a vertex $v$ by $\Gamma(v)$ and its degree by $d(v)$. For two (not necessarily disjoint) sets of vertices $V_1, V_2 \subseteq V$, we let $E(V_1, V_2) \stackrel{\text{def}}{=} \{(v_1, v_2) \in E : v_1 \in V_1, v_2 \in V_2\}$.

Since we shall use the multiplicative Chernoff bound very extensively, we quote it next. Let $\chi_1, \ldots, \chi_m$ be $m$ independent 0/1 valued random variables, where $\Pr[\chi_i = 1] = p$ for every $i$. Then, for every $\eta \in (0, 1]$, the following bounds hold:

$$\Pr\left[\frac{1}{m} \cdot \sum_{i=1}^{m} \chi_i > (1+\eta)p\right] < \exp\left(-\eta^2 pm/3\right)$$

and

$$\Pr\left[\frac{1}{m} \cdot \sum_{i=1}^{m} \chi_i < (1-\eta)p\right] < \exp\left(-\eta^2 pm/2\right).$$

We shall say that an event holds *with high constant probability* if it holds with probability at least $1 - \delta$ for a small constant $\delta$.

Let $\mu$ be a measure defined over graphs, and let $G$ be an unknown graph over $n$ vertices. An algorithm for estimating $\mu(G)$ is given an approximation parameter $\epsilon$, the number of vertices, $n$, and query access to the graph $G$. Here we consider two types of queries. The first are *degree queries*. Namely, for any vertex $v$, the algorithm may ask for the value of $d(v)$. The second are *neighbor queries*. Namely, for any vertex $v$ and for any $1 \le i \le d(v)$, the algorithm may ask for the $i$th neighbor of $v$.[2] We do not make any

[2]Observe that a degree query can be emulated by $\log n$ neighbor queries, but for the sake of the exposition we allow degree queries.

assumption on the order of the neighbors of a vertex. Based on the queries it performs, we ask that the algorithm output an estimate $\hat{\mu}$ of $\mu(G)$ such that with high constant probability (over the random coin flips of the algorithm), $\hat{\mu} = (1 \pm \epsilon) \cdot \mu(G)$, where for $\gamma \in (0,1)$ we use the notation $a = (1 \pm \gamma)b$ to mean that $(1-\gamma)b \leq a \leq (1+\gamma)b$.

**3. An algorithm for approximating the number of length-2 paths.** In this section we describe and analyze an algorithm for estimating the number of length-2 paths (2-stars) in a graph $G$, where we denote this number by $\ell(G)$ (rather than use the slightly more cumbersome notation $\nu_{s2}(G)$). In all that follows we consider undirected simple graphs. Since we introduce quite a lot of notations, we gathered them in Table 3.1 We start by giving the high-level idea behind the algorithm.

**3.1. A high-level description of the algorithm.** Let $\beta = \epsilon/c$, where $c > 1$ is a constant that will be set subsequently, and let $t = \lceil \log_{(1+\beta)} n \rceil$ (so that $t = O(\log n/\epsilon)$). For $i = 0, \ldots, t$, let

TABLE 3.1
*Notations, their meaning, and the location of their exact definition, if appropriate.*

| Notation | Meaning | Exact definition |
|---|---|---|
| $\Gamma(v)$, $d(v)$ | Set of neighbors of $v$, their number | |
| $E(V_1, V_2)$ | $\{(v_1, v_2) \in E : v_1 \in V_1, v_2 \in V_2\}$ | |
| $\epsilon$ | Distance parameter | |
| $\ell(G)$ | Number of length-2 paths | |
| $\tilde{\ell}$ | Given estimate (const. factor) of $\ell(G)$ | |
| $\beta$ | $\epsilon/32$ | |
| $t$ | $\lceil \log_{(1+\beta)} n \rceil$ | |
| $B_i$ | $i$th bucket | Equation (3.1) |
| $\Gamma_i(v)$, $d_i(v)$ | $\Gamma(v) \cap B_i$, $\lvert\Gamma(v) \cap B_i\rvert$ (resp.) | |
| $E_{i,j}$, $E_i$ | $E(B_i, B_j)$, $\bigcup_{j=0}^{t} E_{i,j}$ (resp.) | |
| $B_{i,j,r}$ | Subbucket of $B_i$ | Equation (3.2) |
| $E_{i,j,r}$ | $E(B_{i,j,r}, B_j)$ | |
| $\theta_1$ | Threshold parameter for Algorithm 1 | Step 1 in Algorithm 1 |
| $L$ | Set of indices of large buckets | Step 4 in Algorithm 1 |
| $\theta_2(p)$ | Threshold parameters for Algorithm 2 | Step 1 in Algorithm 2 |
| $LARGE(i,j)$ | $\{r : \lvert B_{i,j,r}\rvert \geq \frac{1}{4}\theta_2(r)\}$ | |
| $p_0$ | Smallest $p$ such that $\frac{1}{4}\theta_2(p+1) \leq n$ | |
| $s^{(p)}$, $g^{(p)}$ | Sample sizes defined in Algorithm 2 | Step 3 in Algorithm 2 |
| $S^{(p)}$, $S_i^{(p)}$, $\hat{S}_{i,j,p}^{(p)}$ | Samples/subsets defined in Algorithm 2 | Step 3 in Algorithm 2 |
| $S_{i,j,r}^{(p)}$ | $S^{(p)} \cap B_{i,j,r}$ | |
| $\gamma_j^{(p)}(v)$ | $S^{(p)} \cap \Gamma_j(v)$ | |
| $\hat{e}_{i,j}$ | Estimate of $\lvert E_{i,j}\rvert$ | Step 4 in Algorithm 2 |
| $\hat{e}_{i,j,r}$ | Contribution of $v \in B_{i,j,r}$ to $\hat{e}_{i,j}$ | Equation (3.19) |
| $\ell^{(\sigma)}(G, \bar{L})$ | Certain numbers of length-2 paths | Definition 1 |
| $SIG$ | Indices of significant buckets | Definition 2 |

$$B_i \stackrel{\text{def}}{=} \{v : d(v) \in ((1+\beta)^{i-1}, (1+\beta)^i]\}. \tag{3.1}$$

We refer to the $B_i$'s as (degree) *buckets*. Note that since degrees are integers, the interval of degrees in each bucket is actually $(\lfloor (1+\beta)^{i-1} \rfloor, \lfloor (1+\beta)^i \rfloor]$, and some buckets are empty. For simplicity we do not use floors unless it has an influence on our analysis, and when we write $\binom{a}{b}$ for $a$ that is not necessarily an integer (e.g., $\binom{(1+\beta)^j}{2}$) then we interpret it as $\binom{\lfloor a \rfloor}{b}$. We also have that $\binom{a}{b} = 0$ for $a < b$ (and, in particular, when $a \leq 0 < b$).

Note that if $n_i$ is the number of nodes with degree $i$ in the graph, then $\ell(G) = \sum_{i=0}^{n-1} n_i \binom{i}{2}$. Suppose that for each bucket $B_i$, we could obtain an estimate, $\hat{b}_i$, such that $(1-\beta)|B_i| \leq \hat{b}_i \leq (1+\beta)|B_i|$. If we let

$$\hat{\ell} = \sum_{i=2}^{t} \hat{b}_i \cdot \binom{(1+\beta)^i}{2},$$

then

$$(1-\beta) \cdot \ell(G) \leq \hat{\ell} \leq (1+\beta)^4 \ell(G)$$

(where we have used the fact that $\binom{(1+\beta)^i}{2} \leq (1+\beta)^3 \cdot \binom{(1+\beta)^{i-1}}{2}$ for $(1+\beta)^{i-1} \geq 2$). If we set $\beta \leq \epsilon/8$, then we get an estimate that is within $(1 \pm \epsilon)$ of the correct value $\ell(G)$. The difficulty is that in order to obtain such an estimate $\hat{b}_i$ of $|B_i|$ in sublinear-time—that is, by sampling—the size of the sample needs to grow with $n/|B_i|$ (so that for small $|B_i|$ the sample is large). Our algorithm indeed takes a sample of vertices, but it uses the sample only to estimate the size of the "large" buckets for an appropriate threshold of "largeness." Using the estimated sizes of the large buckets it can obtain an estimate on the number of length-2 paths whose midpoint belongs to the large buckets.

As noted in the introduction, it is possible that only a small (or even zero) fraction of the length-2 paths have a midpoint that belongs to a large bucket. This implies that we must find a way to estimate the number of length-2 paths whose midpoint is in small buckets (for those small buckets that have a nonnegligible contribution to the total number of length-2 paths).

To this end we do the following. Let $E_{i,j} \stackrel{\text{def}}{=} E(B_i, B_j)$. For each large bucket $B_i$ and small bucket $B_j$ such that the number of length-2 paths whose midpoint is in $B_j$ is nonnegligible, we obtain an estimate $\hat{e}_{i,j}$ to the number $|E_{i,j}|$ of edges between the two buckets. The estimate is such that if $|E_{i,j}|$ is above some threshold, then $\hat{e}_{i,j} = (1 \pm \beta)|E_{i,j}|$, and otherwise $\hat{e}_{i,j}$ is small. Our estimate for the number of length-2 paths whose midpoint is in a small bucket is

$$\frac{1}{2} \sum_{i \in L} \sum_{j \notin L} \hat{e}_{i,j} \cdot ((1+\beta)^j - 1),$$

where $L$ denotes the set of indices of the large buckets. For an illustration, see Figure 3.1. This estimate does not take into account length-2 paths in which no vertices on the path belong to $L$. However, we shall set our threshold of "largeness" so that the number of such paths is negligible. In addition, this estimate takes into account only half of the length-2

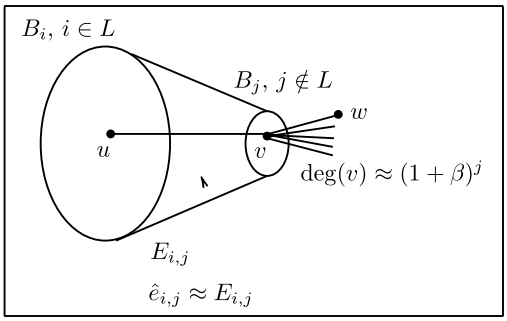


FIG. 3.1. *An illustration for the length-2 paths whose midpoint is in a small bucket.*

paths in which two vertices on the path do not belong to $L$ and one of them is the midpoint. We shall set our threshold of "largeness" so that the number of such paths is also negligible.

One way to estimate $\hat{e}_{i,j}$ (for $i \in L$ and $j \notin L$) is to uniformly select random neighbors of vertices sampled in $B_i$ and check what bucket they belong to. This will indeed give us a good estimate with high probability for a sufficiently large sample. However, the variance in the number of neighbors in $B_j$ that different vertices in $B_i$ have implies that the sample size used by this scheme is significantly larger than necessary. In order to obtain an estimate with a smaller sample, we do the following. For each $i \in L$ and $j \notin L$, we consider partitioning the vertices in $B_i$ that have neighbors in $B_j$ into *subbuckets*. Namely, for $r = 0, \ldots, i$,

$$B_{i,j,r} \stackrel{\text{def}}{=} \{v \in B_i : (1+\beta)^{r-1} < |\Gamma(v) \cap B_j| \leq (1+\beta)^r\}. \tag{3.2}$$

Figure 3.2 illustrates the definition of $B_{i,j,r}$. By the definition of $B_{i,j,r}$,

$$\sum_{r=0}^{i} |B_{i,j,r}| \cdot (1+\beta)^r = (1 \pm \beta) \cdot |E_{i,j}|.$$

Now, if we can obtain good estimates of the sizes of the subsets $|B_{i,j,r}|$, then we get a good estimate for $|E_{i,j}|$. The difficulty is that in order to determine to which subbucket $B_{i,j,r}$ a vertex $v$ belongs, we need to estimate the number of neighbors that it has in $B_j$. This is unlike the case in which we need to determine for a vertex $v$ to which bucket $B_i$ it belongs, where we only need to perform a single degree query. In particular, if

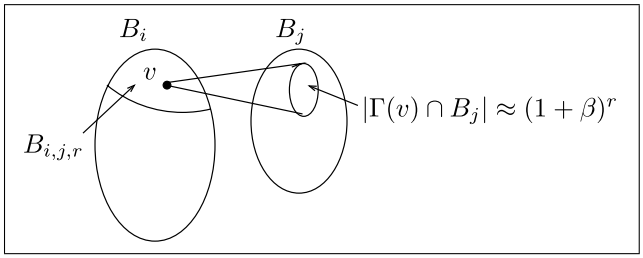


FIG. 3.2. *An illustration for the definition of* $B_{i,j,r}$.

$v \in B_{i,j,0}$—that is, $v$ has a single neighbor in $B_j$—we must query all the neighbors of $v$ in order to determine that it belongs to $B_{i,j,0}$. What works in our favor is the following tradeoff. When $r$ is large, then $|B_{i,j,r}|$ may be relatively small (even if $|E(B_{i,j,r}, B_j)|$ is nonnegligible) so that we need to take a relatively large sample of vertices in order to "hit" $B_{i,j,r}$. However, in order to determine whether a vertex (in $B_i$) belongs to $B_{i,j,r}$ for large $r$, it suffices to take a small sample of its neighbors. On the other hand, when $r$ is relatively small, then $B_{i,j,r}$ must be relatively big (if $|E(B_{i,j,r}, B_j)|$ is nonnegligible). Therefore, it suffices to take a relatively small sample so as to "hit" $B_{i,j,r}$, and then we can afford to perform many neighbor queries from the selected vertices.

We next present our algorithm in detail and then analyze it.

**3.2. The algorithm.** In what follows we assume that we have a rough estimate $\tilde{\ell}$ such that $\frac{1}{2}\ell(G) \leq \tilde{\ell} \leq 2\ell(G)$. We later remove this assumption. Recall that for any two buckets $B_i$ and $B_j$, we use the shorthand $E_{i,j}$ for $E(B_i, B_j)$. The algorithm is given in Algorithm 1.

THEOREM 1. *If $\frac{1}{2}\ell(G) \leq \tilde{\ell} \leq 2\ell(G)$, then with probability at least $2/3$, the output, $\hat{\ell}$, of Algorithm 1 satisfies $\hat{\ell} = (1 \pm \epsilon) \cdot \ell(G)$. The query complexity and running time of the algorithm are $O(\frac{n}{\tilde{\ell}^{1/3}} + \min\{n^{1/2}, \frac{n^{3/2}}{\tilde{\ell}^{1/2}}\}) \cdot \mathrm{poly}(\log n, 1/\epsilon)$.*

Table 3.2 gives the dominant term in the complexity of the algorithm in three different regions of the value of $\ell(G)$ as a function of $n$.

We first prove the second part of Theorem 1, concerning the complexity of the algorithm, and then turn to proving the first part, concerning the quality of the output of the algorithm. We later show how to remove the assumption that the algorithm has an estimate $\tilde{\ell}$ for $\ell(G)$.

ALGORITHM 1. (Estimating the number of length-2 paths for $G = (V, E)$)
*Input:* $\epsilon$ *and* $\tilde{\ell}$.

1. *Let* $\beta \stackrel{\text{def}}{=} \frac{\epsilon}{32}$, $t \stackrel{\text{def}}{=} \left\lceil \log_{(1+\beta)} n \right\rceil$, *and*
$$\theta_1 \stackrel{\text{def}}{=} \frac{\epsilon^{2/3}\tilde{\ell}^{1/3}}{32t^{4/3}} .$$
2. *Uniformly and independently select* $\Theta\left(\frac{n}{\theta_1} \cdot \frac{\log t}{\epsilon^2}\right)$ *vertices from* $V$, *and let* $S$ *denote the multiset of selected vertices (that is, we allow repetitions).*
3. *For* $i = 0, \ldots, t$ *determine* $S_i = S \cap B_i$ *by performing a degree query on every vertex in* $S$.
4. *Let* $L = \left\{ i \ : \ \frac{|S_i|}{|S|} \geq 2\frac{\theta_1}{n} \right\}$.
*If* $\max_{i \in L} \left\{ \binom{(1+\beta)^{i-1}}{2} \cdot \theta_1 \right\} > 4\tilde{\ell}$, *then terminate and return 0.*
5. *For each* $i \in L$ *run Algorithm 2 (see Algorithm 2) to get estimates* $\{\hat{e}_{i,j}\}_{j \notin L}$ *for* $\{|E_{i,j}|\}_{j \notin L}$.
6. *Output*
$$\hat{\ell} = \sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{2} + \sum_{j \notin L} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} \cdot \left((1+\beta)^j - 1\right) .$$

**3.3. Proof of the second part of Theorem 1.** The running time of Algorithm 1 is linear in its query complexity, and hence it suffices to bound the latter. To be precise,

TABLE 3.2
*The query and time complexity of Algorithm* 1.

| $\tilde{\ell}$ | Query and time complexity |
|---|---|
| $\tilde{\ell} \leq n^{3/2}$ | $O(n/\tilde{\ell}^{1/3}) \cdot \text{poly}(\log n, 1/\epsilon)$ |
| $n^{3/2} < \tilde{\ell} \leq n^2$ | $O(n^{1/2}) \cdot \text{poly}(\log n, 1/\epsilon)$ |
| $\tilde{\ell} > n^2$ | $O(n^{3/2}/\tilde{\ell}^{1/2}) \cdot \text{poly}(\log n, 1/\epsilon)$ |

the running time is linear in the upper bound that we give on the query complexity, where this bound sums over the number of queries performed in the different iterations of Algorithm 2. Thus, there may be some recounting since in iteration $p$, queries are performed on vertices in $S^{(p)}$, where $S^p \subset S^{p+1}$.

ALGORITHM 2. (Estimating $\{|\mathrm{E}_{i,j}|\}$ for a given $i \in L$ and all $j \notin L$)
*Input:* $L$, $i \in L$, $\epsilon$ *and* $\tilde{\ell}$.

1. *For each* $0 \leq p \leq i$ *let* $\theta_2(p) \stackrel{\text{def}}{=} \frac{\epsilon^{3/2}\tilde{\ell}^{1/2}}{c_2 t^{5/2}(1+\beta)^{p/2}}$, *where* $c_2$ *is a constant that will be set in the analysis (and where* $t = \left\lceil \log_{(1+\beta)} n \right\rceil$ *for* $\beta = \epsilon/32$*). Let* $p_0$ *be the smallest value of* $p$ *satisfying* $\frac{1}{4}\theta_2(p+1) \leq n$.
2. *For* $p = i$ *down to* $p_0$ *initialize* $\hat{S}^{(p)}_{i,j,p} = \emptyset$.
3. *For* $p = i$ *down to* $p_0$ *do:*
   (a) *Let* $s^{(p)} = \Theta\left(\frac{n}{\theta_2(p)} \cdot \left(\frac{t}{\beta}\right)^2 \log t\right)$, *and let* $g^{(p)} = \Theta\left(\frac{(1+\beta)^{i-p}\log(tn)}{\beta^2}\right)$.
   (b) *Uniformly, independently at random select* $s^{(p)}$ *vertices from* $S^{(p+1)}$ *(where* $S^{(i+1)} = V$*) and let* $S^{(p)}$ *be the multiset of vertices selected.*
   (c) *Determine* $S^{(p)}_i = S^{(p)} \cap B_i$ *by performing a degree query on every vertex in* $S^{(p)}$. *If* $|S^{(p)}_i| < \frac{s^{(p)}}{n} \cdot \frac{1}{4(1+\beta)}\theta_2(p)$, *then go to Step* 4. *Else, if* $|S^{(p)}_i| > \frac{s^{(p)}}{n} \cdot \frac{4\tilde{\ell}}{\binom{(1+\beta)^{i-1}}{2}}$, *then terminate and return* 0.
   (d) *For each* $v \in S^{(p)}_i$ *select (uniformly, independently at random)* $g^{(p)}$ *neighbors of* $v$, *and for each* $j \notin L$ *let* $\gamma^{(p)}_j(v)$ *be the number of these neighbors that belong to* $B_j$. *(If* $g^{(p)} \geq d(v)$, *then consider all neighbors of* $v$.*)*
   (e) *For each* $j \notin L$ *and for each* $v \in S^{(p)}_i \setminus \bigcup_{p'>p} \hat{S}^{(p')}_{i,j,p'}$, *if*

$$\frac{(1+\beta)^{p-1}}{d(v)} < \frac{\gamma^{(p)}_j(v)}{g^{(p)}} \leq \frac{(1+\beta)^p}{d(v)}$$

   *then add* $v$ *to* $\hat{S}^{(p)}_{i,j,p}$.
4. *For each* $j \notin L$ *let* $\hat{e}_{i,j} = \sum_{p=p_0}^{i} \frac{n}{s^{(p)}} \cdot |\hat{S}^{(p)}_{i,j,p}| \cdot (1+\beta)^p$.
5. *Return* $\{\hat{e}_{i,j}\}_{j \notin L}$.

The query complexity of Algorithm 1 is thus bounded by the sum of $\Theta(\frac{n}{\theta_1} \cdot \frac{\log t}{\epsilon^2}) = O(\frac{n}{\tilde{\ell}^{1/3}} \cdot \frac{\log n}{\epsilon^4 \log(1/\epsilon)})$ (the size of the sample selected in step 2 of Algorithm 1) and the number of queries performed in the executions of Algorithm 2. In order

to bound the latter, we first observe that if Algorithm 1 did not terminate in step 4, then

$$\forall i \in L\colon (1+\beta)^i = O\left(\frac{\tilde{\ell}^{1/3} \cdot t^{2/3}}{\epsilon^{1/3}}\right). \tag{3.3}$$

Similarly, if Algorithm 2 did not terminate in any of its executions in step 3c, then, since $\beta = \Theta(\epsilon)$,

$$\forall i \in L \quad \text{and} \quad p_0 \le p \le i\colon |S_i^{(p)}| = O\left(\frac{s^{(p)}}{n} \cdot \frac{\tilde{\ell}}{(1+\beta)^{2i}}\right). \tag{3.4}$$

In addition, it trivially always holds that $|S_i^{(p)}| \le s^{(p)}$. Recall that $p$ runs from $i$ down to $p_0$, where $p_0$ is the smallest value of $p$ satisfying $\frac{1}{4}\theta_2(p+1) \le n$, where $\theta_2(p) \stackrel{\text{def}}{=} \frac{\epsilon^{3/2}\tilde{\ell}^{1/2}}{c_2 t^{5/2}(1+\beta)^{p/2}}$. That is, $p_0 = \lfloor \log_{1+\beta} \frac{\epsilon^3 \tilde{\ell}}{c_2' \cdot t^5 n^2} \rfloor$ for a certain constant $c_2'$. This implies that if $\tilde{\ell} \le \frac{c_2' \cdot t^5}{\epsilon^3} \cdot n^2$, then $p_0 = 0$, and otherwise it may be larger. Therefore, the total number of queries performed in the executions of Algorithm 2 is upper-bounded by

$$\begin{aligned} &\sum_{i \in L} \sum_{p=p_0}^{i} \left( s^{(p)} + \min\left\{ s^{(p)}, \frac{s^{(p)}}{n} \cdot \frac{4\tilde{\ell}}{\binom{(1+\beta)^{i-1}}{2}} \right\} \cdot g^{(p)} \right) \\ &\le \sum_{i \in L} i \cdot s^{(i)} + \sum_{i \in L} \sum_{p=p_0}^{i} \min\left\{ s^{(p)}, \frac{s^{(p)}}{n} \cdot \frac{4\tilde{\ell}}{\binom{(1+\beta)^{i-1}}{2}} \right\} \cdot g^{(p)}. \end{aligned} \tag{3.5}$$

For the first summand in (3.5), we apply (3.3), the definitions of $s^{(i)}$ and $\theta_2(i)$, the fact that $\beta = \Theta(\epsilon)$, and the fact that $i \le t$, and we get

$$\begin{aligned} \sum_{i \in L} i \cdot s^{(i)} &\le \sum_{i \in L} t \cdot O\left(\frac{n}{\theta_2(i)} \cdot \left(\frac{t}{\beta}\right)^2 \log\, t\right) \\ &= \sum_{i \in L} t \cdot O\left(\frac{n \cdot t^{9/2} \log\, t \cdot (1+\beta)^{i/2}}{\epsilon^{7/2}\tilde{\ell}^{1/2}}\right) \\ &= O\left(\frac{n \cdot t^{13/2} \log\, t \cdot \left(\frac{\tilde{\ell}^{1/3} \cdot t^{2/3}}{\epsilon^{1/3}}\right)^{1/2}}{\epsilon^{7/2}\tilde{\ell}^{1/2}}\right) \\ &= O\left(\frac{n}{\tilde{\ell}^{1/3}} \cdot \frac{t^7 \log\, t}{\epsilon^4}\right). \end{aligned} \tag{3.6}$$

Turning to the second summand in (3.5) and again using the definitions of $s^{(p)}$, $\theta_2(p)$ as well as $g^{(p)}$ and $\beta = \Theta(\epsilon)$, we get

$$\sum_{i\in L}\sum_{p=p_0}^{i}\min\left\{s^{(p)},\frac{s^{(p)}}{n}\cdot\frac{4\tilde{\ell}}{\binom{(1+\beta)^{i-1}}{2}}\right\}\cdot g^{(p)}$$

$$=\sum_{i\in L}\sum_{p=p_0}^{i}O\left(\min\left\{\frac{n}{\theta_2(p)}\cdot\left(\frac{t}{\beta}\right)^2\log\,t,\frac{1}{\theta_2(p)}\cdot\left(\frac{t}{\beta}\right)^2\log\,t\cdot\frac{4\tilde{\ell}}{\binom{(1+\beta)^{i-1}}{2}}\right\}\right.$$

$$\left.\cdot\frac{(1+\beta)^{i-p}\,\log(tn)}{\beta^2}\right)$$

$$=\sum_{i\in L}\sum_{p=p_0}^{i}O\Bigg(\min\left\{\frac{n\cdot(1+\beta)^{p/2}}{\tilde{\ell}^{1/2}}\cdot\frac{t^{9/2}\log\,t}{\epsilon^{7/2}},\frac{\tilde{\ell}^{1/2}}{(1+\beta)^{2i-p/2}}\cdot\frac{t^{9/2}\log\,t}{\epsilon^{7/2}}\right\}$$

$$\cdot\frac{(1+\beta)^{i-p}\,\log(tn)}{\beta^2}\Bigg)$$

$$=\sum_{i\in L}\sum_{p=p_0}^{i}O\Bigg(\min\left\{\frac{n\cdot(1+\beta)^{i-p/2}}{\tilde{\ell}^{1/2}},\frac{\tilde{\ell}^{1/2}}{(1+\beta)^{i+p/2}}\right\}\cdot\frac{t^{9/2}\log\,t\log(tn)}{\epsilon^{11/2}}\Bigg)$$

$$=\sum_{i\in L}O\Bigg(\min\left\{\frac{n\cdot(1+\beta)^{i}}{\tilde{\ell}^{1/2}},\frac{\tilde{\ell}^{1/2}}{(1+\beta)^{i}}\right\}\cdot\frac{t^{9/2}\log\,t\log(tn)}{\epsilon^{11/2}}\Bigg)$$

$$\cdot(1+\beta)^{-p_0/2}\cdot\sum_{k=0}^{i-p_0}(1+\beta)^{-k/2}. \tag{3.7}$$

In order to bound the expression in (3.7), we first note that if $(1+\beta)^i\le\frac{\tilde{\ell}^{1/2}}{n^{1/2}}$, then $\frac{n\cdot(1+\beta)^i}{\tilde{\ell}^{1/2}}\le n^{1/2}$, while if $(1+\beta)^i\ge\frac{\tilde{\ell}^{1/2}}{n^{1/2}}$, then $\frac{\tilde{\ell}^{1/2}}{(1+\beta)^i}\le n^{1/2}$ as well. Since $(1+\beta)^{-p_0/2}=1$ and $\sum_{k=0}^{i-p_0}(1+\beta)^{-k/2}=O(1/\beta)$, if $p_0=0$, then the right-hand side of (3.7) is upper-bounded by

$$O\left(n^{1/2}\cdot\frac{t^{11/2}\log\,t\log(tn)}{\epsilon^{13/2}}\right). \tag{3.8}$$

If $p_0>0$, then the bound in (3.8) should be multiplied by $(1+\beta)^{-p_0/2}$. By definition of $p_0$, we have that $(1+\beta)^{-p_0/2}=O(\frac{t^{5/2}n}{\epsilon^{3/2}\tilde{\ell}^{1/2}})$, and so we get the (tighter) bound

$$O\left(n^{1/2}\cdot\frac{t^{11/2}\log\,t\log(tn)}{\epsilon^{13/2}}\right)\cdot(1+\beta)^{-p_0/2}=O\left(\frac{n^{3/2}}{\tilde{\ell}^{1/2}}\cdot\frac{t^8\log\,t\log(tn)}{\epsilon^{10}}\right). \tag{3.9}$$

The total number of queries performed in the executions of Algorithm 2 is hence upper-bounded by

$$O\left(\frac{n}{\tilde{\ell}^{1/3}}+\min\left\{n^{1/2},\frac{n^{3/2}}{\tilde{\ell}^{1/2}}\right\}\right)\cdot\mathrm{poly}(\log\,n,1/\epsilon). \tag{3.10}$$

**3.4. Proof of the first part of Theorem 1.** In what follows we claim that certain events occur with high constant probability, and in some cases we claim that they hold with larger probability (e.g., $1-\frac{1}{\mathrm{poly}(t)}$). In all cases the statement holds for sufficiently large constants in the $\Theta(\cdot)$ notations for the sample sizes used by the algorithm. Recall that the algorithm takes a sample of size $\Theta(\frac{n}{\theta_1}\cdot\frac{\log t}{\epsilon^2})$, and $S$ denotes the multiset of examples it got.

LEMMA 1. *With high constant probability, for every $i$ such that $|B_i|\geq\theta_1$, it holds that $\frac{|S_i|}{|S|}=(1\pm\frac{\epsilon}{8})\frac{|B_i|}{n}$, and for every $i$ such that $|B_i|<\theta_1$, it holds that $\frac{|S_i|}{|S|}<2\frac{\theta_1}{n}$.*

*Proof.* The proof follows by applying the multiplicative Chernoff bound and a union bound. Since the expected size of $S_i$ is $\frac{|B_i|}{n}\cdot|S|$, it holds that if $|B_i|\geq\theta_1$, then

$$\begin{aligned}\Pr\left[\frac{|S_i|}{|S|}>\left(1+\frac{\epsilon}{8}\right)\frac{|B_i|}{n}\right]&<\exp\left(-\frac{\epsilon^2|B_i|\cdot|S|}{192\cdot n}\right)\\&<\exp\left(-\Omega\left(\frac{\epsilon^2\theta_1\cdot n\cdot\log t}{n\cdot\theta_1\epsilon^2}\right)\right)=\frac{1}{\mathrm{poly}(t)}.\end{aligned}\tag{3.11}$$

In the same manner we get that $\Pr[\frac{|S_i|}{|S|}<(1-\frac{\epsilon}{8})\frac{|B_i|}{n}]<\frac{1}{\mathrm{poly}(t)}$. On the other hand, if $|B_i|<\theta_1$, then the expected size of $S_i$ is upper-bounded by $\frac{\theta_1}{n}\cdot|S|$, so we get that

$$\Pr\left[\frac{|S_i|}{|S|}>2\frac{\theta_1}{n}\right]<\exp\left(-\frac{\theta_1}{n}\frac{|S|}{3}\right)=\exp\left(-\Omega\left(\frac{\epsilon^2\theta_1\cdot n\cdot\log t}{n\cdot\theta_1\epsilon^2}\right)\right)=\frac{1}{\mathrm{poly}(t)},$$

and the lemma follows. ∎

As a direct corollary of Lemma 1 and the definition of $L$ in Algorithm 1, we get the following.

COROLLARY 2. *With high constant probability, for every $i\in L$, we have that $\frac{|S_i|}{|S|}=(1\pm\frac{\epsilon}{8})\frac{|B_i|}{n}$, and for every $i\notin L$, we have that $|B_i|<4\theta_1$.*

The first part of Corollary 2 implies that (with high constant probability) the estimate $\sum_{i\in L}n\cdot\frac{|S_i|}{|S|}\cdot\binom{(1+\beta)^i}{2}$ is close to the actual number of length-2 paths whose midpoint belongs to a bucket $B_i$ such that $i\in L$. It also implies that Algorithm 1 does not terminate in step 4 (with high constant probability). To verify this, first observe that since $\tilde{\ell}\geq\frac{1}{2}\ell(G)$, for every $1\leq i\leq t$ we have that $|B_i|\cdot\binom{(1+\beta)^{i-1}}{2}\leq\ell(G)\leq2\tilde{\ell}$. By the definition of $L$, for every $i\in L$ we have that $\frac{|S_i|}{|S|}\geq\frac{2\theta_1}{n}$. If the termination condition holds—that is, there exists an index $i\in L$ for which $2\theta_1\cdot\binom{(1+\beta)^{i-1}}{2}>4\tilde{\ell}$—then $n\cdot\frac{|S_i|}{|S|}\cdot\binom{(1+\beta)^{i-1}}{2}>4\tilde{\ell}$ for that index $i$. But by Corollary 2, with high constant probability, for every $i\in L$ we have that $\frac{|S_i|}{|S|}=(1\pm\frac{\epsilon}{8})\frac{|B_i|}{n}$, which implies that $|B_i|\cdot\binom{(1+\beta)^{i-1}}{2}>2\tilde{\ell}\geq\ell(G)$, and we reach a contradiction.

The remainder of the analysis deals with the quality of the estimate for the number of length-2 paths in $G$ whose midpoint is not in $L$.

DEFINITION 1. *For $j\notin L$ and $\sigma\in\{1,2,3\}$, let $\ell_j^{(\sigma)}(G,\bar{L})$ denote the number of length-2 paths in $G$ whose midpoint belongs to $B_j$ and such that the number of vertices on the*

*path that belong to $B_k$ for $k \notin L$ (including $j$) is $\sigma$. For $\sigma \in \{1,2,3\}$, let $\ell^{(\sigma)}(G,\bar{L}) = \sum_{j \notin L} \ell_j^{(\sigma)}(G,\bar{L})$ and for every $j \notin L$, let $\ell_j(G,\bar{L}) = \sum_{\sigma=1}^{3} \ell_j^{(\sigma)}(G,\bar{L})$.*

We first observe that with high constant probability both $\ell^{(3)}(G,\bar{L})$ and $\ell^{(2)}(G,\bar{L})$ are relatively small.

LEMMA 3. *With high constant probability, $\ell^{(3)}(G,\bar{L}) \leq \frac{\epsilon}{4}\ell(G)$ and $\ell^{(2)}(G,\bar{L}) \leq \frac{\epsilon}{4}\ell(G)$.*

*Proof.* First observe that by the second part of Corollary 2 and the definition of $\theta_1$ we have that with high constant probability,

$$\sum_{j \notin L} |B_j| < \frac{1}{8t^{1/3}} \epsilon^{2/3} \tilde{\ell}^{1/3}.$$

By our assumption that $\tilde{\ell} \leq 2\ell(G)$,

$$\ell^{(3)}(G,\bar{L}) \leq \binom{\sum_{j \notin L} |B_j|}{3} \leq \binom{\epsilon^{2/3}\tilde{\ell}^{1/3}/(8t^{1/3})}{3} < \frac{\epsilon}{8}\tilde{\ell} \leq \frac{\epsilon}{4}\ell(G).$$

In order to bound $\ell^{(2)}(G,\bar{L})$ we observe that since the total number of length-2 paths is $\ell(G)$, for every bucket $B_j$ we have that $\binom{(1+\beta)^{j-1}+1}{2} \leq \ell(G)/|B_j|$, and so

$$(1+\beta)^j \leq 2\frac{\ell^{1/2}(G)}{|B_j|^{1/2}}.$$

Therefore,

$$\begin{aligned}
\ell^{(2)}(G,\bar{L}) &\leq \sum_{j \notin L} |B_j| \cdot (1+\beta)^j \cdot \sum_{k \notin L} |B_k| \\
&\leq \frac{\epsilon^{2/3}\tilde{\ell}^{1/3}}{4t^{1/3}} \cdot \sum_{j \notin L} (\ell^{1/2}(G) \cdot |B_j|^{1/2}) \\
&\leq \frac{\epsilon^{2/3}\tilde{\ell}^{1/3}}{4t^{1/3}} \cdot \ell^{1/2}(G) \cdot t \cdot \frac{\epsilon^{1/3}\tilde{\ell}^{1/6}}{2\sqrt{2}t^{2/3}} \\
&< \frac{\epsilon}{4}\ell(G),
\end{aligned}$$

and the proof is completed. ∎

Lemma 3 implies that in order to obtain a good estimate on the number of length-2 paths whose midpoint belongs to small buckets, it suffices to get a good estimate on the number of such paths that have at least one endpoint in a large bucket.[3] We next define the notion of significant buckets for buckets $B_j$ such that $j \notin L$. Roughly speaking, nonsignificant small buckets are buckets that we can ignore, or, more precisely, we can undercount the number of edges between vertices in them and vertices in large buckets.

DEFINITION 2 (significant small buckets). *For every $j \notin L$, we say that $j$ is* significant *if*

[3]The assertion follows from the first part of Lemma 3, which bounds $\ell^{(3)}(G,\bar{L})$. The reason that we also need a bound on $\ell^{(2)}(G,\bar{L})$ will be made clear subsequently.

$$|B_j| \cdot \binom{(1+\beta)^j}{2} \geq \frac{\epsilon}{c_3 t}\tilde{\ell},$$

*where $c_3$ is a constant that will be set in the analysis. We denote the set of indices of significant buckets $B_j$ (where $j \notin L$) by* SIG.

Note that by the definition of SIG,

$$\sum_{j \notin L, j \notin SIG} \ell_j(G, \bar{L}) < \frac{\epsilon}{c_3}\tilde{\ell} \leq \frac{2\epsilon}{c_3}\ell(G). \tag{3.12}$$

Let

$$E_j \stackrel{\text{def}}{=} \bigcup_{k=0}^{t} E_{j,k}, \tag{3.13}$$

and recall that $\theta_2(r) \stackrel{\text{def}}{=} \frac{\epsilon^{3/2}\tilde{\ell}^{1/2}}{c_2 t^{5/2}(1+\beta)^{r/2}}$. We have the following lemma concerning significant buckets.

LEMMA 4. *If $j \in SIG$, then for every $r$ such that $|B_{i,j,r}| > 0$ for some $i$, we have that*

$$|E_j| \geq \frac{(c_2/c_3^{1/2})t^2}{\epsilon}\theta_2(r) \cdot (1+\beta)^r.$$

The implication of Lemma 4 is roughly the following. Consider any $j \in SIG$ and a nonempty subbucket $B_{i,j,r}$. Recall that by the definition of $B_{i,j,r}$ the number of edges between $B_{i,j,r}$ and $B_j$ is approximately $|B_{i,j,r}| \cdot (1+\beta)^r$. Suppose that $B_{i,j,r}$ is small, and, in particular, that it is smaller than $\theta_2(r)$. Then the number of edges between $B_{i,j,r}$ and $B_j$ as a fraction of all the edges incident to $B_j$—that is, $E_j$—is $O(\epsilon/t^2)$, which is negligible. This means that we may underestimate the size of such small subbuckets without incurring a large error.

*Proof.* Since $j$ is significant,

$$(1+\beta)^j > \sqrt{\frac{2\epsilon\tilde{\ell}}{c_3 t|B_j|}}. \tag{3.14}$$

Since the graph contains no multiple edges, $|B_j| \geq (1+\beta)^r$ for each $r$ such that $B_{i,j,r}$ is not empty. Therefore,

$$|E_j| \geq |B_j| \cdot (1+\beta)^{j-1} \tag{3.15}$$

$$\geq \frac{1}{1+\beta}\sqrt{\frac{2\epsilon\tilde{\ell}|B_j|}{c_3 t}} \tag{3.16}$$

$$\geq \frac{1}{c_3^{1/2}t^{1/2}} \cdot \epsilon^{1/2}\tilde{\ell}^{1/2}(1+\beta)^{r/2} \tag{3.17}$$

$$\geq \frac{(c_2/c_3^{1/2})t^2}{\epsilon}\theta_2(r) \cdot (1+\beta)^r, \tag{3.18}$$

where (3.15) follows from the definitions of $|E_j|$ and $|B_j|$, (3.16) follows from (3.14), (3.17) follows from the lower bound just stated on $|B_j|$, and (3.18) follows from the definition of $\theta_2(r) = \frac{\epsilon^{3/2}\tilde{\ell}^{1/2}}{c_2 t^{5/2}(1+\beta)^{r/2}}$), and the proof is completed. ∎

Armed with Lemmas 3 and 4 we now turn to analyzing Algorithm 2. We start with a high-level discussion and then turn to the precise details.

**The high-level structure of the analysis of Algorithm 2.** Recall that the algorithm works iteratively as follows. It first takes (uniformly, independently, at random) a sample $S^i$ from $V$, and in further iterations $0 \leq p < i$ the sample $S^{(p)}$ is selected (uniformly, independently, at random) from $S^{(p+1)}$. Since the same vertex may be selected more than once, the $S^{(p)}$'s may actually be multisets. For each $p$, the algorithm tries to estimate $|B_{i,j,p}|$ by deciding for each vertex $v \in S^{(p)} \cap B_i$ whether it belongs to $B_{i,j,p}$. This is done by sampling from the neighbors of $v$ and checking what fraction of its neighbors belong to $B_j$. If the fraction is within the interval $[\frac{(1+\beta)^{p-1}}{d(v)}, \frac{(1+\beta)^p}{d(v)}]$, then $v$ is assumed to belong to $B_{i,j,p}$ and is put in a corresponding subset $\hat{S}^{(p)}_{i,j,p}$.

The difficulty is that this estimate of the fraction of neighbors in $B_j$ may deviate somewhat from its expected value. As a result, vertices that belong to $B_{i,j,p}$ may not be deemed so. In particular, consider a vertex $v$ such that $\Gamma_j(v)$ (the number of neighbors that $v$ has in $B_j$) is close to the lower bound of $(1+\beta)^{p-1}$ (or the upper bound $(1+\beta)^p$). It is then possible that the fraction of neighbors of $v$ in $B_j$ that belong to the sample falls below $\frac{(1+\beta)^{p-1}}{d(v)}$ (or above $\frac{(1+\beta)^p}{d(v)}$, respectively), in which case $v$ will not be added to $\hat{S}^{(p)}_{i,j,p}$. Similarly, vertices that do not belong to $B_{i,j,p}$ but have a number of neighbors in $B_j$ that is close to $(1+\beta)^{p-1}$ or $(1+\beta)^p$—that is, vertices that belong to $B_{i,j,p-1}$ or $B_{i,j,p+1}$—may be added to $\hat{S}^{(p)}_{i,j,p}$.

If the size of the sample $S^{(p)}$ was the same for all $p$, then the above would not really be a difficulty: we could take a single sample $S = S^i$ and work iteratively from $p = i$ down to $p = 0$. For each $p$, we would consider only those vertices $v$ that were not yet added to $\hat{S}^{(p')}_{i,j,p'}$ for $p' > p$ and decide whether to add $v$ to $\hat{S}^{(p)}_{i,j,p}$. By the above discussion, for every $r$ and every $v \in B_{i,j,r}$, the vertex $v$ would be put in $\hat{S}^{(r+1)}_{i,j,r+1}$, $\hat{S}^{(r)}_{i,j,r}$, or $\hat{S}^{(r-1)}_{i,j,r-1}$. The algorithm would then output, as an estimate for $|E_{i,j}|$, the sum over all $0 \leq p \leq i$ of $\frac{n}{|S|} \cdot |\hat{S}^{(p)}_{i,j,p}|(1+\beta)^r$. If $S \cap B_{i,j,r}$ is close to its expected size for each $r$, then the deviation of the final estimate from $|E_{i,j}|$ can be easily bounded.

However, as $p$ decreases from $i$ to 0, we need to use a smaller sample $S^{(p)}$. Recall that a smaller sample suffices since $\theta_2(p)$ increases when $p$ decreases, and it is necessary to use a smaller sample because the cost of estimating the number of neighbors in $B_j$ increases as $p$ decreases. Thus, in each iteration $p$, the new, smaller sample, $S^{(p)}$, is selected from the sample $S^{(p+1)}$ of the previous iteration. What we would like to ensure is that (1) the size of each subset $S^{(p)}_{i,j,r} \stackrel{\text{def}}{=} S^{(p)} \cap B_{i,j,r}$ is close to its expectation, and (2) if some fraction of $S^{(p+1)}_{i,j,r}$ was added to $\hat{S}^{(p+1)}_{i,j,p+1}$ for $r = p+1$ or $r = p$, then in the new sample $S^{(p)}$, the size of $S^{(p)} \cap (S^{(p+1)}_{i,j,r} \setminus \hat{S}^{(p+1)}_{i,j,p+1})$ is close to its expectation. Here, when we say "close to its expectation," we mean up to a multiplicative factor of $(1 \pm O(\epsilon))$. This should be the case unless the expected value is below some threshold (which is determined by $\theta_2(r)$). If the expected value is below the threshold, then it suffices that we do not get a significant overestimate. To understand the idea for why this suffices, see the discussion following Lemma 4. Further details follow.

Recall that $s^{(p)}$ denotes the size of the sample $S^{(p)}$, where $s^{(p)} = \Theta(\frac{n}{\theta_2(p)} \cdot (\frac{t}{\beta})^2 \log\, t)$. The next lemma establishes that by our choice of $s^{(p)}$, if a fixed subset of $S^{(p+1)}$ is sufficiently large, then the number of its vertices that are selected in $S^{(p)}$ is close to the expected value, and if it is small, then few of its vertices will appear in $S^{(p)}$. Lemma 5 follows directly by applying a multiplicative Chernoff bound (and will be applied to various subsets of the samples $S^{(p)}$).

LEMMA 5. *For any fixed choice of $\tilde{S}^{(p+1)} \subseteq S^{(p+1)}$, if $\frac{|\tilde{S}^{(p+1)}|}{s^{(p+1)}} \geq \frac{\theta_2(p)}{8n}$, then, with probability at least $1 - \frac{1}{32t^4}$,*

$$\frac{1}{1+\frac{\beta}{2(i+1)}} \cdot \frac{|\tilde{S}^{(p+1)}|}{s^{(p+1)}} \leq \frac{|S^{(p)} \cap \tilde{S}^{(p+1)}|}{s^{(p+1)}} \leq \left(1 + \frac{\beta}{2(i+1)}\right) \cdot \frac{|\tilde{S}^{(p+1)}|}{s^{(p+1)}},$$

*and if $\frac{|\tilde{S}^{(p+1)}|}{s^{(p+1)}} < \frac{\theta_2(p)}{8n}$, then with probability at least $1 - \frac{1}{32t^4}$,*

$$\frac{|S^{(p)} \cap \tilde{S}^{(p+1)}|}{s^{(p)}} < \left(1 + \frac{\beta}{2(i+1)}\right) \cdot \frac{\theta_2(p)}{8n}.$$

Let $S_i^{(p)} \stackrel{\text{def}}{=} S^{(p)} \cap B_i$ and let $S_{i,j,r}^{(p)} \stackrel{\text{def}}{=} S^{(p)} \cap B_{i,j,r}$. (Note that $S_i^{i+1} = B_i$ and $S_{i,j,r}^{i+1} = B_{i,j,r}$.) Since $\theta_2(p)$ is monotonically decreasing with $p$ (so that $s^{(p)}$ is monotonically increasing with $r$), and because $(1 + \frac{\beta}{2(i+1)})^{i+1} \leq 1 + \beta$, Lemma 5 implies the next corollary.

COROLLARY 6. *With high constant probability, for every $i \in L$ and $j \notin L$, and for every $r$ such that $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$, we have that for every $r - 1 \leq p \leq i$,*

$$\left(\frac{1}{1+\frac{\beta}{2(i+1)}}\right)^{i-p+1} \cdot \frac{|B_{i,j,r}|}{n} \leq \frac{|S_{i,j,r}^{(p)}|}{s^{(p)}} \leq \left(1 + \frac{\beta}{2(i+1)}\right)^{i-p+1} \cdot \frac{|B_{i,j,r}|}{n}.$$

*On the other hand, if $|B_{i,j,r}| < \frac{1}{4}\theta_2(r)$, then*

$$\frac{|S_{i,j,r}^{(p)}|}{s^{(p)}} < (1+\beta) \cdot \frac{\theta_2(r)}{4n}$$

*for every $p$.*

Lemma 5 also implies that with high constant probability, Algorithm 2 does not terminate in step 3c. Recall that the algorithm terminates in step 3c if $n \cdot \frac{|S_i^{(p)}|}{s^{(p)}} \geq \frac{1}{4(1+\beta)}\theta_2(p)$ and $n \cdot \frac{|S_i^{(p)}|}{s^{(p)}} \cdot \binom{(1+\beta)^{i-1}}{2} > 4\tilde{\ell}$. By Lemma 5, with probability at least $1 - \frac{1}{32t^2}$, for every $i$ and $p$, if $|B_i| < \frac{1}{6}\theta_2(p)$, then $n \cdot \frac{|S_i^{(p)}|}{s^{(p)}} \leq (1+\beta)\frac{1}{6}\theta_2(p)$, and if $|B_i| \geq \frac{1}{6}\theta_2(p)$, then $n \cdot \frac{|S_i^{(p)}|}{s^{(p)}} \leq (1+\beta)|B_i|$. Assuming this is in fact the case, if $|B_i| < \frac{1}{6}\theta_2(p)$, then $n \cdot \frac{|S_i^{(p)}|}{s^{(p)}} < \frac{1}{4(1+\beta)}\theta_2(p)$, so that the algorithm will not terminate. On the other hand, if $|B_i| \geq \frac{1}{6}\theta_2(p)$, then

$$n \cdot \frac{|S_i^{(p)}|}{s^{(p)}} \cdot \binom{(1+\beta)^{i-1}}{2} \leq (1+\beta)|B_i| \cdot \binom{(1+\beta)^{i-1}}{2} \leq (1+\beta)\ell(G) < 4\tilde{\ell},$$

so that the algorithm will not terminate in this case as well.

The next lemma deals with the estimates we get for the number of neighbors that a vertex in $B_i$ has in $B_j$, and it too follows from the multiplicative Chernoff bound. In the lemma and what follows, we shall use the notations $\Gamma_j(v) \stackrel{\text{def}}{=} \Gamma(v) \cap B_j$ and $d_j(v) \stackrel{\text{def}}{=} |\Gamma_j(v)|$.

LEMMA 7. *Let $i \in L$, $j \notin L$ and for each $0 \leq p \leq i$, let $g^{(p)} = \Theta(\frac{(1+\beta)^{i-p} \cdot \log(t \cdot n)}{\beta^2})$. For any $r \geq p-1$ and for any fixed choice of a vertex $v \in S_{i,j,r}^{(p)}$, if we take a sample of size $g^{(p)}$ of neighbors of $v$ and let $\gamma_j^{(p)}(v)$ be the number of neighbors in the sample that belong to $\Gamma_j(v)$, then with probability at least $1 - \frac{1}{16 \cdot t \cdot n}$,*

$$\frac{1}{1+\beta} \cdot \frac{d_j(v)}{d(v)} \leq \frac{\gamma_j^{(p)}(v)}{g^{(p)}} \leq (1+\beta) \cdot \frac{d_j(v)}{d(v)}.$$

*In addition, for each $r \leq p-2$ and $v \in S_{i,j,r}^{(p)}$, with probability at least $1 - \frac{1}{16 \cdot t \cdot n}$,*

$$\frac{\gamma_j^{(p)}(v)}{g^{(p)}} < \frac{(1+\beta)^{p-1}}{d(v)}.$$

The next lemma is central to our analysis. Ideally we would have liked each vertex in the sample to be added to its "correct" subset. That is, if $v \in S_{i,j,r}^{(r)}$ $(= S^{(r)} \cap B_{i,j,r})$, then ideally it should be added to $\hat{S}_{i,j,r}^{(r)}$. However, since the decision concerning whether to add a vertex to a particular subset depends on sampling its neighbors and estimating the number of neighbors that it has in $B_j$, we cannot ensure that it will be added to precisely the right subset. However, we can ensure (with high probability) that it will not be added to a subset $\hat{S}_{i,j,p}^{(p)}$ for $p$ that differ significantly from $r$.

LEMMA 8. *With high constant probability, for every $i \in L$, $j \notin L$, $0 \leq r \leq i$, and $v \in B_{i,j,r}$ such that $v$ is selected in the initial sample $S^i$, the vertex $v$ may belong to $\hat{S}_{i,j,r+1}^{(r+1)}$, $\hat{S}_{i,j,r}^{(r)}$, or $\hat{S}_{i,j,r-1}^{(r-1)}$, but not to any other $\hat{S}_{i,j,r'}^{(r')}$. In other words, $\hat{S}_{i,j,r}^{(r)} \subseteq B_{i,j,r+1} \cup B_{i,j,r} \cup B_{i,j,r-1}$.*

*Proof.* By the definition of $B_{i,j,r}$, if $v \in B_{i,j,r}$, then $(1+\beta)^{r-1} < d_j(v) \leq (1+\beta)^r$. By Lemma 7, for each $p \leq r+1$ with probability at least $1 - \frac{1}{16 \cdot t \cdot n}$,

$$\frac{1}{1+\beta} \cdot \frac{(1+\beta)^{r-1}}{d(v)} < \frac{\gamma_j^{(p)}(v)}{g^{(p)}} \leq (1+\beta) \cdot \frac{(1+\beta)^{r+1}}{d(v)}.$$

That is, for each $p \leq r+1$ and, in particular, for $r-1 \leq p \leq r+1$,

$$\frac{(1+\beta)^{r-2}}{d(v)} < \frac{\gamma_j^{(p)}(v)}{g^{(p)}} \leq \frac{(1+\beta)^{r+2}}{d(v)}.$$

On the other hand, for $p \geq r+2$, with probability at least $1 - \frac{1}{16 \cdot t \cdot n}$,

$$\frac{\gamma_j^{(p)}(v)}{g^{(p)}} < \frac{(1+\beta)^{p-1}}{d(v)}.$$

By taking a union bound over all vertices $v$, and for each $v \in B_{i,j,r}$ over all $0 \le p \le i$, this implies the following:

1. For $r+2 \le p \le i$, no vertex in $S_{i,j,r}^{(p)}$ is added to $\hat{S}_{i,j,r}^{(p)}$.
2. For $r-1 \le p \le r+1$, the following holds: If a vertex $v$ belongs to $S_{i,j,r}^{(r+1)}$, then it may be added to $\hat{S}_{i,j,r+1}^{(r+1)}$, and if not, then it may be added to $\hat{S}_{i,j,r}^{(r)}$ (assuming $v \in S^{(r)}$). If it was added to neither of the two subsets and it is selected in $S^{(r-1)}$, then it is added to $\hat{S}_{i,j,r-1}^{(r-1)}$ (and it will not be added to $\hat{S}_{i,j,r}^{(p)}$ for any $p < r-1$).

The proof is completed. ∎

We are now ready to prove that the estimates $\hat{e}_{i,j}$ computed by Algorithm 2 are essentially close to the corresponding values of $|E_{i,j}|$. Recall that *SIG* denotes the set of all significant indices (as defined in Definition 2) and that $E_j \stackrel{\text{def}}{=} \bigcup_{k=0}^{t} E_{j,k}$.

LEMMA 9. *For an appropriate choice of $c_2$ (in the definition of $\theta_2(\cdot)$ in step 1 of Algorithm 2) and of $c_3$ (in Definition 2), with high constant probability, for all $j \notin L$, if $j \in SIG$, then*

$$\left(1-\frac{\epsilon}{8}\right)\sum_{i\in L}|E_{i,j}| - \frac{\epsilon}{16}|E_j| \le \sum_{i\in L}\hat{e}_{i,j} \le \left(1+\frac{\epsilon}{4}\right)|E_j|,$$

*and if $j \notin SIG$, then*

$$\sum_{i\in L}\frac{1}{2}\,\hat{e}_{i,j}\cdot((1+\beta)^j - 1) \le \frac{\epsilon}{4t}\ell(G).$$

*Proof.* Recall that

$$\hat{e}_{i,j} = \sum_{p=p_0}^{i}\frac{n}{s^{(p)}}\cdot|\hat{S}_{i,j,p}^{(p)}|\cdot(1+\beta)^p.$$

By Lemma 8, with high constant probability, for every $i$, $j$, $r$ such that $r \ge p_0+1$, the contribution of vertices in $B_{i,j,r}$ to this sum is

$$\begin{aligned} &n\cdot\left(\frac{|\hat{S}_{i,j,r+1}^{(r+1)}\cap B_{i,j,r}|}{s^{(r+1)}}\cdot(1+\beta)^{r+1} + \frac{|\hat{S}_{i,j,r}^{(r)}\cap B_{i,j,r}|}{s^{(r)}}\cdot(1+\beta)^r\right)\\ &+n\cdot\left(\frac{|\hat{S}_{i,j,r-1}^{(r-1)}\cap B_{i,j,r}|}{s^{(r-1)}}\cdot(1+\beta)^{r-1}\right). \end{aligned} \tag{3.19}$$

Assume from now on that this is in fact true and denote this sum by $\hat{e}_{i,j,r}$. Consider first the case that $|B_{i,j,r}| < \frac{1}{4}\theta_2(r)$.

*Claim* 10. With high constant probability, for every $i$, $j$, $r$ such that $|B_{i,j,r}| < \frac{1}{4}\theta_2(r)$, if $j \in SIG$, then

$$\hat{e}_{i,j,r} \le \frac{\epsilon}{c_4 t^2}|E_j| \quad \text{for } c_4 = c_2/c_3^{1/2},$$

and if $j \notin SIG$, then

$$\hat{e}_{i,j,r} \cdot ((1+\beta)^j - 1) \leq \frac{\epsilon}{c_5 t^3} \ell(G) \quad \text{for } c_5 = c_2 c_3^{1/2}/2.$$

*Proof.* By Corollary 6, with high constant probability, for every $i$, $j$, $r$, if $|B_{i,j,r}| < \frac{1}{4}\theta_2(r)$, then $\frac{|S^{(p)}_{i,j,r}|}{s^{(p)}} < (1+\beta) \cdot \frac{\theta_2(r)}{4n}$ for every $p$. Assuming this is in fact the case, we have that

$$\hat{e}_{i,j,r} \leq \frac{3}{4} \cdot \theta_2(r) \cdot (1+\beta)^{r+2}. \tag{3.20}$$

If $j \in SIG$, then by Lemma 4 we have that $|E_j| \geq \frac{(c_2/c_3^{1/2})t^2}{\epsilon} \theta_2(r) \cdot (1+\beta)^r$. Therefore,

$$\hat{e}_{i,j,r} \leq \frac{\epsilon}{c_4 t^2} |E_j| \tag{3.21}$$

for $c_4 = c_2/c_3^{1/2}$ (using $\beta \leq 1/32$).

If $j \notin SIG$, then $(1+\beta)^j \leq \frac{2}{c_3^{1/2}} \left(\frac{\epsilon\tilde{\ell}}{t|B_j|}\right)^{1/2}$. Using the fact that $(1+\beta)^r \leq |B_j|$ (because there are no multiple edges) and by the definition of $\theta_2(r)$, we get that

$$\begin{aligned} \hat{e}_{i,j,r} \cdot ((1+\beta)^j - 1) &\leq \frac{1}{c_2 t^{5/2}} \cdot \epsilon^{3/2} \cdot \tilde{\ell}^{1/2} \cdot |B_j|^{1/2} \cdot \frac{2}{c_3^{1/2}} \left(\frac{\epsilon\tilde{\ell}}{t|B_j|}\right)^{1/2} \\ &\leq \frac{\epsilon}{c_5 t^3} \tilde{\ell} \leq \frac{\epsilon}{c_5 t^3} \ell(G) \end{aligned} \tag{3.22}$$

for $c_5 = c_2 c_3^{1/2}/2$. (Claim 10) ∎

We now turn to the case that $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$.

*Claim* 11. With high constant probability, for every $i$, $j$, $r$ such that $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$, if $j \in SIG$, then

$$(1+\beta)^{-3}|E_{i,j,r}| - \frac{\epsilon}{c_4' t^2}|E_j| \leq \hat{e}_{i,j,r} \leq (1+\beta)^3 |E_{i,j,r}| + \frac{\epsilon}{c_4' t^2}|E_j|$$

for $c_4' = c_2/(2c_3^{1/2})$, and for $j \notin SIG$,

$$\hat{e}_{i,j,r} \cdot ((1+\beta)^j - 1) \leq (1+\beta)^3 |E_{i,j,r}| \cdot ((1+\beta)^j - 1) + \frac{\epsilon}{c_5' t^3} \tilde{\ell}$$

for $c_5' = c_2 c_3^{1/2}/4$.

*Proof.* By Corollary 6, with high constant probability, for every $i$, $j$, $r$, if $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$, then for every $r-1 \leq p \leq i$,

$$\left(\frac{1}{1+\frac{\beta}{2(i+1)}}\right)^{i-p+1} \cdot \frac{|B_{i,j,r}|}{n} \leq \frac{|S^{(p)}_{i,j,r}|}{s^{(p)}} \leq \left(1+\frac{\beta}{2(i+1)}\right)^{i-p+1} \cdot \frac{|B_{i,j,r}|}{n}. \tag{3.23}$$

Assume from this point on that this is in fact the case. Fixing such a choice of $i$, $j$, $r$, let

$$\tilde{S}^{(r+1)}_{i,j,r} \stackrel{\text{def}}{=} S^{(r+1)}_{i,j,r} \setminus \hat{S}^{(r+1)}_{i,j,r+1} \quad \text{and} \quad \tilde{S}^{(r)}_{i,j,r} \stackrel{\text{def}}{=} S^{(r)}_{i,j,r} \setminus (\hat{S}^{(r+1)}_{i,j,r+1} \cup \hat{S}^{(r)}_{i,j,r}).$$

That is, $\tilde{S}^{(r+1)}_{i,j,r}$ is the subset of vertices in $S^{(r+1)}_{i,j,r}$ $(= S^{(r+1)} \cap B_{i,j,r})$ that were not added to $\hat{S}^{(r+1)}_{i,j,r+1}$, and $\tilde{S}^{(r)}_{i,j,r}$ is the subset of vertices in $S^{(r)}_{i,j,r}$ $(= S^{(r)} \cap B_{i,j,r})$ that were added to neither $\hat{S}^{(r+1)}_{i,j,r+1}$ nor to $\hat{S}^{(r)}_{i,j,r}$. Let

$$\alpha_1 \stackrel{\text{def}}{=} \frac{|\tilde{S}^{(r+1)}_{i,j,r}|}{|S^{(r+1)}_{i,j,r}|} \quad \text{and} \quad \alpha_2 \stackrel{\text{def}}{=} \frac{|\tilde{S}^{(r)}_{i,j,r}|}{|\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}.$$

Since $S^{(r+1)}_{i,j,r} = (\hat{S}^{(r+1)}_{i,j,r} \cap B_{i,j,r}) \cup \tilde{S}^{(r+1)}_{i,j,r}$ (where the two subsets on the right-hand side are disjoint), according to the definition of $\alpha_1$ we have that $|\hat{S}^{(r+1)}_{i,j,r} \cap B_{i,j,r}| = (1-\alpha_1)|S^{(r+1)}_{i,j,r}|$. By (3.23),

$$\frac{|\hat{S}^{(r+1)}_{i,j,r+1} \cap B_{i,j,r}|}{s^{(r+1)}} = \frac{(1-\alpha_1) \cdot |S^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \le (1+\beta)(1-\alpha_1)\frac{|B_{i,j,r}|}{n},$$

and similarly

$$\frac{|\hat{S}^{(r+1)}_{i,j,r+1} \cap B_{i,j,r}|}{s^{(r+1)}} \ge (1-\beta)(1-\alpha_1)\frac{|B_{i,j,r}|}{n}.$$

**The case of large $\alpha_1$ and $\alpha_2$.** In order to obtain bounds on the second and third terms in (3.19), assume first that both

$$\frac{|\tilde{S}^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \ge \frac{\theta_2(r)}{4n} \quad \text{and} \quad \frac{|\tilde{S}^{(r)}_{i,j,r}|}{s^{(r)}} \ge \frac{\theta_2(r)}{4n}.$$

That is,

$$\frac{\alpha_1 \cdot |S^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \ge \frac{\theta_2(r)}{4n} \quad \text{and} \quad \frac{\alpha_2 \cdot |\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}} \ge \frac{\theta_2(r)}{4n}.$$

Under this assumption, by Lemma 5, with probability at least $1 - \frac{1}{32t^4}$,

$$\frac{|\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}} \le \left(1 + \frac{\beta}{2i}\right)\frac{|\tilde{S}^{(r+1)}_{i,j,r}|}{s^{(r+1)}} = \left(1 + \frac{\beta}{2i}\right)\frac{\alpha_1 \cdot |S^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \tag{3.24}$$

and

$$\frac{|\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}} \ge \left(1 - \frac{\beta}{2i}\right)\frac{\alpha_1 \cdot |S^{(r+1)}_{i,j,r}|}{s^{(r+1)}}. \tag{3.25}$$

Similarly, with probability at least $1 - \frac{1}{32t^4}$,

$$\frac{|\tilde{S}^{(r)}_{i,j,r} \cap S^{(r-1)}|}{s^{(r-1)}} \le \left(1 + \frac{\beta}{2i}\right)\frac{|\tilde{S}^{(r)}_{i,j,r}|}{s^{(r)}} = \left(1 + \frac{\beta}{2i}\right)\frac{\alpha_2 \cdot |\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}} \tag{3.26}$$

and

$$\frac{|\tilde{S}^{(r)}_{i,j,r} \cap S^{(r-1)}|}{s^{(r-1)}} \geq \left(1 - \frac{\beta}{2i}\right) \frac{\alpha_2 \cdot |\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}}. \tag{3.27}$$

Assume that (3.24)–(3.27) indeed hold. Observe that $\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)} = (\hat{S}^{(r)}_{i,j,r} \cap B_{i,j,r}) \cup \tilde{S}^{(r)}_{i,j,r}$ (where the two subsets on the right-hand side are disjoint), so that by the definition of $\alpha_2$ we have that $|\hat{S}^{(r)}_{i,j,r} \cap B_{i,j,r}| = (1-\alpha_2)|\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|$. By (3.23) and (3.24),

$$\begin{aligned}
\frac{|\hat{S}^{(r)}_{i,j,r} \cap B_{i,j,r}|}{s^{(r)}} &= (1-\alpha_2)\frac{|\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}} \\
&\leq \left(1 + \frac{\beta}{2i}\right)(1-\alpha_2)\alpha_1 \cdot \frac{|S^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \\
&\leq (1+\beta)(1-\alpha_2)\alpha_1 \cdot \frac{B_{i,j,r}}{n}, \qquad (3.28)
\end{aligned}$$

and similarly (by (3.23) and (3.25)),

$$\frac{|\hat{S}^{(r)}_{i,j,r} \cap B_{i,j,r}|}{s^{(r)}} \geq (1-\beta)(1-\alpha_2)\alpha_1 \cdot \frac{B_{i,j,r}}{n}.$$

Finally, by our assumption (which holds with high probability) that sampled vertices in $B_{i,j,r}$ are added to $\hat{S}^{(r+1)}_{i,j,r+1}$, $\hat{S}^{(r)}_{i,j,r}$, or $\hat{S}^{(r-1)}_{i,j,r-1}$, all vertices in $\tilde{S}^{(r)}_{i,j,r} \cap S^{(r-1)}$ are added to $\hat{S}^{(r-1)}_{i,j,r-1}$. Therefore, by (3.23), (3.24), and (3.26) (and the definitions of $\alpha_1$ and $\alpha_2$),

$$\begin{aligned}
\frac{|\hat{S}^{(r-1)}_{i,j,r-1} \cap B_{i,j,r}|}{s^{(r-1)}} &= \frac{|\tilde{S}^{(r)}_{i,j,r} \cap S^{(r-1)}|}{s^{(r-1)}} \\
&\leq \left(1 + \frac{\beta}{2i}\right)\frac{|\tilde{S}^{(r)}_{i,j,r}|}{s^{(r)}} \\
&= \left(1 + \frac{\beta}{2i}\right)\alpha_2 \frac{|\tilde{S}^{(r+1)}_{i,j,r} \cap S^{(r)}|}{s^{(r)}} \\
&\leq \left(1 + \frac{\beta}{2i}\right)^2 \alpha_2 \frac{|\tilde{S}^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \\
&= \left(1 + \frac{\beta}{2i}\right)^2 \alpha_2\alpha_1 \frac{|S^{(r+1)}_{i,j,r}|}{s^{(r+1)}} \\
&\leq (1+\beta)\alpha_2\alpha_1 \cdot \frac{B_{i,j,r}}{n}. \qquad (3.29)
\end{aligned}$$

Similarly (by (3.23), (3.25), and (3.27)),

$$\frac{|\hat{S}^{(r-1)}_{i,j,r-1} \cap B_{i,j,r}|}{s^{(r-1)}} \geq (1-\beta)\alpha_2\alpha_1 \cdot \frac{B_{i,j,r}}{n}. \tag{3.30}$$

**The case of small $\alpha_1$ or small $\alpha_2$.** The bounds in (3.28)–(3.30) were obtained for the case that both $\alpha_1$ and $\alpha_2$ are above certain thresholds. If

$\alpha_1 \cdot |S_{i,j,r}^{(r+1)}|/s^{(r+1)} < \theta_2(r)/(4n)$—that is, $|\tilde{S}_{i,j,r}^{(r+1)}|/s^{(r+1)} < \theta_2(r)/(4n)$—then by Lemma 5, with probability at least $1 - \frac{1}{16t^4}$,

$$\frac{|\tilde{S}_{i,j,r}^{(r+1)} \cap S^{(r)}|}{s^{(r)}} \leq (1+\beta)\frac{\theta_2(r)}{4n}$$

and

$$\frac{|\tilde{S}_{i,j,r}^{(r)} \cap S^{(r-1)}|}{s^{(r-1)}} \leq (1+\beta)\frac{\theta_2(r)}{4n}$$

as well. Similarly, if $\alpha_2 \cdot |\tilde{S}_{i,j,r}^{(r+1)} \cap S^{(r)}|/s^{(r)} < \theta_2(r)/(4n)$—that is, $|\tilde{S}_{i,j,r}^{(r)}|/s^{(r)} < \theta_2(r)/(4n)$—then with probability at least $1 - \frac{1}{32t^4}$,

$$\frac{|\tilde{S}_{i,j,r}^{(r)} \cap S^{(r-1)}|}{s^{(r-1)}} \leq (1+\beta)\frac{\theta_2(r)}{4n}.$$

By combining all the bounds above we get that (for $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$)

$$\begin{aligned}\hat{e}_{i,j,r} &\leq (1+\beta)\Big((1-\alpha_1)|B_{i,j,r}|(1+\beta)^{r+1} + \alpha_1(1-\alpha_2)|B_{i,j,r}|(1+\beta)^r \\ &\quad + \alpha_1\alpha_2|B_{i,j,r}|(1+\beta)^{r-1}\Big) + 2\theta_2(r)(1+\beta)^r \\ &\leq |B_{i,j,r}|(1+\beta)^{r+2} + 2\theta_2(r)(1+\beta)^r \qquad (3.31)\end{aligned}$$

and

$$\hat{e}_{i,j,r} \geq |B_{i,j,r}|(1+\beta)^{r-2} - \theta_2(r)(1+\beta)^{r+1}. \qquad (3.32)$$

Similar to what we have shown for the case that $|B_{i,j,r}| < \frac{1}{4}\theta_2(r)$ (see (3.20)–(3.22)), if we let $E_{i,j,r} \stackrel{\text{def}}{=} E(B_{i,j,r}, B_j)$, then we get that for $j \in SIG$,

$$(1+\beta)^{-3}|E_{i,j,r}| - \frac{\epsilon}{c_4' t^2}|E_j| \leq \hat{e}_{i,j,r} \leq (1+\beta)^3|E_{i,j,r}| + \frac{\epsilon}{c_4' t^2}|E_j|, \qquad (3.33)$$

and for $j \notin SIG$,

$$\hat{e}_{i,j,r} \cdot ((1+\beta)^j - 1) \leq (1+\beta)^3|E_{i,j,r}| \cdot ((1+\beta)^j - 1) + \frac{\epsilon}{c_5' t^3}\tilde{\ell} \qquad (3.34)$$

for $c_4' = c_2/(2c_3^{1/2})$ and for $c_5' = c_2 c_3^{1/2}/4$. (Claim 11) ∎

Let $LARGE(i,j)$ denote the subset of indices $r$ for which $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$. By Claim 10 (for the case that $|B_{i,j,r}| < \frac{1}{4}\theta_2(r)$) and Claim 11 (for the case that $|B_{i,j,r}| \geq \frac{1}{4}\theta_2(r)$), and by taking a union bound, we get that the following bounds hold with high constant probability. First, for every $j \in SIG$,

$$\begin{aligned}
\sum_{i\in L}\hat{e}_{i,j} &= \sum_{i\in L}\sum_{p=p_0}^{i}\frac{n}{s^{(p)}}\cdot|\hat{S}^{(p)}_{i,j,p}|\cdot(1+\beta)^p \\
&\le \sum_{i\in L}\sum_{r\in LARGE(i,j)}\hat{e}_{i,j,r}+\sum_{i\in L}\sum_{r\notin LARGE(i,j)}\hat{e}_{i,j,r} \\
&\le \left(1+\frac{\epsilon}{8}\right)\sum_{i\in L}\sum_{r\in LARGE(i,j)}|E_{i,j,r}|+\frac{\epsilon}{c_4'}|E_j|+\frac{\epsilon}{c_4}|E_j| \\
&\le \left(1+\frac{\epsilon}{8}\right)\sum_{i\in L}|E_{i,j}|+\frac{\epsilon}{c_4'}|E_j|+\frac{\epsilon}{c_4}|E_j| \\
&\le \left(1+\frac{\epsilon}{4}\right)|E_j|,
\end{aligned}$$

where the last inequality holds conditioned on $c_4$ and $c_4'$ (which are functions of $c_2$ and $c_3$) being sufficiently large (and, in particular, holds for any $c_3 \ge 1$ and $c_2 \ge 32\cdot c_3^{1/2}$). Recall that $p_0$ is the smallest value of $p$ satisfying $\frac{1}{4}\theta_2(p+1)\le n$. Since $|B_{i,j,r}|\le n$ for every $i$, $j$, $r$ while $|B_{i,j,r}|\ge\frac{1}{4}\theta_2(r)$ for every $r\in LARGE(i,j)$, we have that $r\ge p_0+1$ for every $r\in LARGE(i,j)$. Therefore,

$$\begin{aligned}
\sum_{i\in L}\hat{e}_{i,j} &\ge \sum_{i\in L}\sum_{r\in LARGE(i,j)}\hat{e}_{i,j,r} \\
&\ge \left(1-\frac{\epsilon}{8}\right)\sum_{i\in L}\sum_{r\in LARGE(i,j)}|E_{i,j,r}|-\frac{\epsilon}{c_4'}|E_j| \\
&\ge \left(1-\frac{\epsilon}{8}\right)\sum_{i\in L}\left(\sum_{r=0}^{i}|E_{i,j,r}|-\sum_{r\notin LARGE(i,j)}|E_{i,j,r}|\right)-\frac{\epsilon}{c_4'}|E_j| \\
&\ge \left(1-\frac{\epsilon}{8}\right)\sum_{i\in L}|E_{i,j}|-\epsilon\left(\frac{1}{c_4}+\frac{1}{c_4'}\right)\cdot|E_j| \\
&\ge \left(1-\frac{\epsilon}{8}\right)\sum_{i\in L}|E_{i,j}|-\frac{\epsilon}{16}\cdot|E_j|,
\end{aligned}$$

where the last inequality holds for sufficiently large $c_4$ and $c_4'$ (and, in particular, whenever $c_3\ge 1$ and $c_2\ge 64\cdot c_3^{1/2}$). On the other hand, for $j\notin SIG$,

$$\begin{aligned}
\sum_{i\in L}\frac{1}{2}\hat{e}_{i,j}\cdot((1+\beta)^j-1) &= \sum_{i\in L}\frac{1}{2}\sum_{r\in LARGE(i,j)}\hat{e}_{i,j,r}\cdot((1+\beta)^j-1) \\
&\quad+\sum_{i\in L}\frac{1}{2}\sum_{r\notin LARGE(i,j)}\hat{e}_{i,j,r}\cdot((1+\beta)^j-1) \\
&\le \left(1+\frac{\epsilon}{8}\right)\frac{1}{2}\sum_{i\in L}\sum_{r\in LARGE(i,j)}|E_{i,j,r}|\cdot((1+\beta)^j-1) \\
&\quad+\frac{\epsilon}{c_5' t}\ell(G)+\frac{\epsilon}{c_5 t}\ell(G) \\
&\le \left(1+\frac{\epsilon}{8}\right)\frac{\epsilon}{c_3 t}\tilde{\ell}+\frac{\epsilon}{t}\left(\frac{1}{c_5'}+\frac{1}{c_5}\right)\ell(G) \qquad (3.35)
\end{aligned}$$

$$< \frac{\epsilon}{4t}\ell(G), \tag{3.36}$$

where in (3.35) we built on the definition of significant buckets and the last equation holds for sufficiently large $c_3$, $c_5$, and $c_5'$ (and, in particular, for any choice of $c_3 \geq 32$ and $c_2 \geq 64/c_3^{1/2}$). By taking $c_3 \geq 32$ and $c_2 \geq 64 \cdot c_3^{1/2}$, the proof of Lemma 9 is completed. ∎

**Putting it all together: Proving the first part of Theorem 1.** Recall that

$$\hat{\ell} = \sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{2} + \sum_{j \notin L} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} \cdot ((1+\beta)^j - 1). \tag{3.37}$$

Let $\ell(G, L)$ denote the number of length-2 paths in $G$ whose midpoint belongs to a bucket $B_i$ such that $i \in L$, and let $\ell(G, \bar{L})$ denote the number of length-2 paths whose midpoint belongs to a bucket $B_j$ such that $j \notin L$ (so that $\ell(G, L) + \ell(G, \bar{L}) = \ell(G)$). By the first part of Corollary 2 (and the setting of $\beta$), we have that with high constant probability

$$\sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{2} = \left(1 \pm \frac{\epsilon}{4}\right)\ell(G, L). \tag{3.38}$$

Turning to the second summand in (3.37), by Lemma 9,

$$\begin{aligned}
\sum_{j \notin L} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} \cdot ((1+\beta)^j - 1) &= \sum_{j \notin L, j \in SIG} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} \cdot ((1+\beta)^j - 1) \\
&\quad + \sum_{j \notin L, j \notin SIG} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} \cdot ((1+\beta)^j - 1) \\
&\leq \sum_{j \notin L, j \in SIG} \frac{1}{2} \cdot \left(1 + \frac{\epsilon}{4}\right)|E_j| \cdot ((1+\beta)^j - 1) + \frac{\epsilon}{4}\ell(G) \\
&\leq \left(1 + \frac{\epsilon}{4}\right) \cdot \sum_{j \notin L} \frac{1}{2}|E_j| \cdot ((1+\beta)^j - 1) + \frac{\epsilon}{4}\ell(G) \\
&\leq \left(1 + \frac{\epsilon}{2}\right)\ell(G, \bar{L}) + \frac{\epsilon}{4}\ell(G). \qquad (3.39)
\end{aligned}$$

In the other direction, recall that $\ell^{(\sigma)}(G, \bar{L}) = \sum_{j \notin L} \ell_j^{(\sigma)}(G, \bar{L})$, where for $j \notin L$ and $\sigma \in \{1, 2, 3\}$, we let $\ell_j^{(\sigma)}(G, \bar{L})$ denote the number of length-2 paths whose midpoint belongs to $B_j$ and such that the number of vertices on the path that belong to $B_k$ for $k \notin L$ (including $j$) is $\sigma$,

$$\sum_{j\notin L}\frac{1}{2}\sum_{i\in L}\hat{e}_{i,j}\cdot(1+\beta)^j \geq \sum_{j\notin L, j\in SIG}\frac{1}{2}\left(\sum_{i\in L}\left(1-\frac{\epsilon}{8}\right)|E_{i,j}|-\frac{\epsilon}{16}|E_j|\right)\cdot((1+\beta)^j-1)$$

$$\geq \sum_{j\notin L}\frac{1}{2}\sum_{i\in L}\left(1-\frac{\epsilon}{8}\right)|E_{i,j}|\cdot((1+\beta)^j-1)-\frac{(1+\beta)\epsilon}{16}\ell(G)$$

$$-\sum_{j\notin L, j\notin SIG}\frac{1}{2}\sum_{i\in L}\left(1-\frac{\epsilon}{8}\right)|E_{i,j}|\cdot((1+\beta)^j-1)$$

$$\geq \left(1-\frac{\epsilon}{8}\right)(\ell^{(1)}(G,\bar{L})+\frac{1}{2}\ell^{(2)}(G,\bar{L}))-\frac{\epsilon}{4}\ell(G) \tag{3.40}$$

$$\geq \left(1-\frac{\epsilon}{8}\right)(\ell^{(1)}(G,\bar{L})+\ell^{(2)}(G,\bar{L})+\ell^{(3)}(G,\bar{L}))$$

$$-\frac{1}{2}\ell^{(2)}(G,\bar{L})-\ell^{(3)}(G,\bar{L})-\frac{\epsilon}{4}\ell(G)$$

$$\geq \left(1-\frac{\epsilon}{8}\right)\ell(G,\bar{L})-\frac{3\epsilon}{4}\ell(G), \tag{3.41}$$

where in (3.40) we used (3.12) (based on the definition of $SIG$ and taking $c_3 \geq 32$ as it was set previously), and in the last inequality we applied Lemma 3. By combining (3.38), (3.39), and (3.41), we get that $\hat{\ell} = (1\pm\epsilon)\ell(G)$ with high constant probability.

**3.5. Removing the assumption on $\tilde{\ell}$.** Our analysis builds on the assumption that $\frac{1}{2}\ell(G) \leq \tilde{\ell} \leq 2\ell(G)$. In order to get rid of this assumption, we observe that if we run Algorithm 1 with $\tilde{\ell} > 2\ell(G)$, then our analysis implies that with high constant probability $\hat{\ell} \leq (1+\frac{\epsilon}{2})\ell(G)+\frac{\epsilon}{8}\tilde{\ell}$. This is true because (1) the algorithm still obtains (with high probability) an estimate of $\ell(G,L)$ that does not overestimate $\ell(G,L)$ by more than a factor of $(1+\frac{\epsilon}{4})$, (2) for the number of length-2 paths whose midpoint is in a bucket $B_j$, where $j\notin L$ and $j\in SIG$, the approximation factor is at most $(1+\frac{\epsilon}{2})$, and (3) the additional error caused by overestimating the number of length-2 paths whose midpoint is in a bucket $B_j$, where $j\notin L$ and $j\notin SIG$, is at most $\frac{\epsilon}{8}\tilde{\ell}$.

Suppose we run Algorithm 1 with $\tilde{\ell} > 2\ell(G)$. Then with high constant probability $\hat{\ell} < (\frac{1}{2}+\frac{\epsilon}{2})\tilde{\ell}$. On the other hand, if we run Algorithm 1 with $\frac{1}{2}\ell(G) \leq \tilde{\ell} < \ell(G)$, then with high constant probability, $\hat{\ell} \geq (1-\epsilon)\ell(G) > (1-\epsilon)\tilde{\ell}$, which is greater than $(\frac{1}{2}+\frac{\epsilon}{2})\tilde{\ell}$ for every $\epsilon < 1/3$.

Therefore, we do the following. Starting from $\tilde{\ell} = n\cdot\binom{n}{2}$, we repeatedly call a slight variant of Algorithm 1 with our current estimate $\tilde{\ell}$. The variant is that we increase all sample sizes by a factor of $\Theta(\log\log n)$ so as to reduce the failure probability of each execution to $O(1/\log n)$, and we run the algorithm with $\epsilon = \min\{\epsilon, 1/4\}$. In each execution we reduce the previous value of $\tilde{\ell}$ by a factor of 2, and stop once $\hat{\ell} > (1-\epsilon)\tilde{\ell}$, at which point we output $\hat{\ell}$. By the above discussion, with high constant probability we do not stop before $\tilde{\ell}$ goes below $2\ell(G)$, and conditioned on this, with high probability $(1-O(1/\log n))$ we do stop once $\frac{1}{2}\ell(G) \leq \tilde{\ell} < \ell(G)$ (or possibly, one iteration earlier, when $\ell(G) \leq \tilde{\ell} < 2\ell(G)$) with $\hat{\ell} = (1\pm\epsilon)\ell(G)$.

Since there is a nonzero probability that the algorithm does not stop when $\frac{1}{2}\ell(G) \leq \tilde{\ell} < \ell(G)$, we next bound the expected running time of the algorithm. The total running time of all executions until $\frac{1}{2}\ell(G) \leq \tilde{\ell} < \ell(G)$ is $O(\frac{n}{\ell(G)^{1/3}} + \min\{n^{1/2}, \frac{n^{3/2}}{\ell(G)^{1/2}}\})\cdot\mathrm{poly}(\log n, 1/\epsilon)$. Once $\tilde{\ell} < \frac{1}{2}\ell(G)$, the algorithm may terminate

in step 4 of Algorithm 1 or in step 3c of Algorithm 2, but if it does not, then the probability that $\hat{\ell} \leq (1-\epsilon)\tilde{\ell}$ in any particular execution is upper-bounded by $O(1/\log n)$. Since the executions are independent, the expected running time is $O(\frac{n}{\ell(G)^{1/3}} + \min\{n^{1/2}, \frac{n^{3/2}}{\ell(G)^{1/2}}\}) \cdot \text{poly}(\log n, 1/\epsilon)$.

We thus have the following theorem.

THEOREM 2. *With probability at least* $2/3$*, the aforementioned algorithm, which uses Algorithm* 1 *as a subroutine, returns an estimates* $\hat{\ell}$ *that satisfies* $\hat{\ell} = (1 \pm \epsilon) \cdot \ell(G)$*. The expected query complexity and running time of the algorithm are* $O(\frac{n}{(\ell(G))^{1/3}} + \min\{n^{1/2}, \frac{n^{3/2}}{(\ell(G))^{1/2}}\}) \cdot \text{poly}(\log n, 1/\epsilon)$.

**4. Lower bounds for approximating the number of length-2 paths.** In the next theorem we state three lower bounds that together match our upper bound in terms of the dependence on $n$ and $\ell(G)$. In what follows, when we refer to a multiplicative approximation algorithm for the number of length-2 paths, we mean an algorithm that outputs an estimate $\hat{\ell}$ that with high probability satisfies $\ell(G)/C \leq \hat{\ell} \leq C\ell(G)$ for some (predetermined) approximation factor $C$ (where $C$ may depend on the size of the graph). If $C$ is a constant, then the algorithm is a constant-factor approximation algorithm.

THEOREM 3.

1. *Any multiplicative approximation algorithm for* $\ell(G)$ *must perform* $\Omega(\frac{n}{\ell^{1/3}(G)})$ *queries.*
2. *Any constant-factor approximation algorithm for* $\ell(G)$ *must perform* $\Omega(\sqrt{n})$ *queries when the number of length-2 paths is* $O(n^2)$*.*
3. *Any constant-factor approximation algorithm for* $\ell(G)$ *must perform* $\Omega(\frac{n^{3/2}}{\ell^{1/2}(G)})$ *queries when the number of length-2 paths is* $\Omega(n^2)$*.*

**4.1. Proof of item 1 in Theorem 3.** To establish the first item in Theorem 3, we show that every $n$ and for every value of $\ell$, there exists a family of $n$-vertex graphs for which the following holds. For each graph $G$ in the family, we have that $\ell(G) = \Theta(\ell)$, but it is not possible to distinguish (with high constant probability) by making $o(n/\ell^{1/3})$ queries, between a random graph in the family and the empty graph (for which $\ell(G) = 0$). Each graph in the family simply consists of a clique of size $b = \lceil \ell^{1/3} \rceil$ and an independent set of size $n - b$. The number of length-2 paths in the graph is $b \cdot \binom{b-1}{2} = \Theta(\ell)$. However, in order to distinguish between a random graph in the family and the empty graph, it is necessary to perform a query on a vertex in the clique. The probability of hitting such a vertex in $o(\frac{n}{\ell^{1/3}(G)})$ queries is $o(1)$.

**4.2. Proof of item 2 in Theorem 3.** Since we have already established in item 1 in Theorem 3 that there is a lower bound of $\Omega(\frac{n}{\ell^{1/3}(G)})$, and since for $\ell(G) \leq n^{3/2}$, we have that $\frac{n}{\ell^{1/3}(G)} \geq n^{1/2}$, we may consider the case that $\ell(G) > n^{3/2} > n$. To establish item 2 in Theorem 3 we show that for every $n$, every constant $c$, and every $n < \ell < (n/2c)^2$ there exist two families of $n$-vertex graphs for which the following holds. In both families the number of length-2 paths is $\Theta(\ell)$, but in one family this number is a factor $c$ larger than in the other family. However, it is not possible to distinguish with high constant probability between a graph selected randomly in one family and a graph selected randomly in the other family using $o(\sqrt{n})$ queries. We first present two families that include some graphs with multiple edges and self-loops, and then modify the construction to obtain simple graphs.

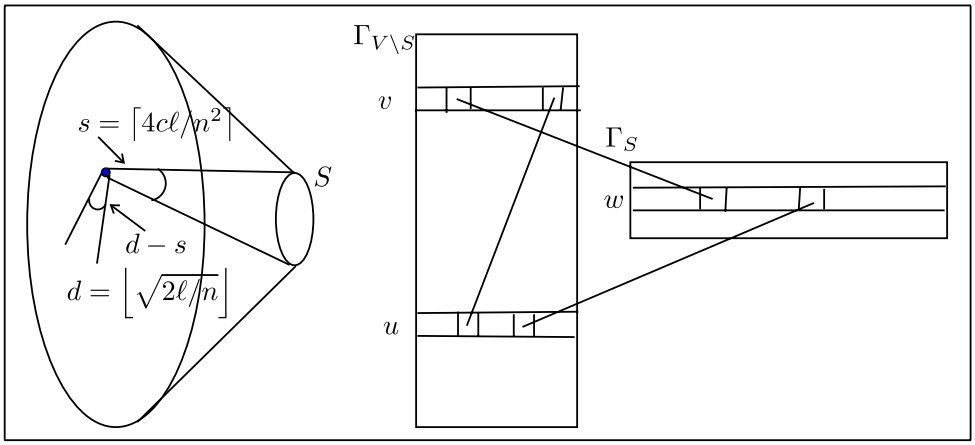


FIG. 4.1. *An illustration for the proof of item* 2 *in Theorem* 3. *On the left-hand side is a graph in* $\mathcal{G}_2$, *and on the right-hand side are the corresponding neighborhood tables,* $\Gamma_{V\setminus S}$ *and* $\Gamma_S$. *Each row in* $\Gamma_{V\setminus S}$ *corresponds to a vertex in* $V \setminus S$ *and each row in* $\Gamma_S$ *corresponds to a vertex in* $S$. *A connecting line between a pair of entries in the two tables indicates that there is an edge between the two corresponding vertices.*

**The graph families.** In the first family, denoted $\mathcal{G}_1$, each graph is a union of $d = \lfloor\sqrt{2\ell/n}\rfloor$ matchings. Thus, each vertex has degree $d = \lfloor\sqrt{2\ell/n}\rfloor$ and

$$\ell(G) = n \cdot \binom{d}{2} < \ell.$$

A random graph in $\mathcal{G}_1$ is determined by simply selecting $d$ random matchings. In the second family, denoted $\mathcal{G}_2$, each graph is determined as follows. There is a small subset, $S$, of $c$ vertices, where each vertex in $S$ has degree $d' = \lceil\sqrt{2\ell}\rceil + 1$, and each vertex in $V \setminus S$ has degree $d = \lfloor\sqrt{2\ell/n}\rfloor$ (like all vertices in the graph in $\mathcal{G}_1$). If we view each vertex in $S$ as having $d'$ *ports* (one for each incident edge) and each vertex in $V \setminus S$ as having $d$ ports, then a graph is the family $\mathcal{G}_2$ is defined by a perfect matching between the $(n-c)\cdot d + c \cdot d'$ ports (we assume this number is even, otherwise, $d$ and $d'$ can be slightly modified). For an illustration, see the left-hand side of Figure 4.1. Thus,

$$\ell(G) > c \cdot \binom{d'}{2} = c \cdot \binom{\lceil\sqrt{2\ell}\rceil + 1}{2} > c\ell.$$

**Processes that construct graphs in the families.** In order to show that it is hard to distinguish between graphs selected randomly from the two families in $o(\sqrt{n})$ queries, we follow [GR02], [KKR04] and define two processes, $\mathcal{P}_1$ and $\mathcal{P}_2$, that interact with an approximation algorithm $\mathcal{A}$. The process $\mathcal{P}_1$ answers the queries of $\mathcal{A}$ while constructing a random graph in $\mathcal{G}_1$, and the process $\mathcal{P}_2$ answers the queries of $\mathcal{A}$ while constructing a random graph in $\mathcal{G}_2$. We consider the distributions over the respective *query-answer histories*, $\langle (q_1, a_1), \ldots, (q_t, a_t)\rangle$, and show that for histories of length $o(\sqrt{n})$, the distributions are very close, implying that $\mathcal{A}$ must have a high failure probability if it performs only $o(\sqrt{n})$ queries. Details follow.

For simplicity we assume that for every vertex that appears in either a neighbor query or an answer to such a query, both processed give the degree of the vertex "for free," so there is no need for degree queries. We also assume that an answer $u$ to a neighbor query $(v, i)$ comes with the label $i'$ of the edge from $u$'s side of the edge.

Clearly any lower bound under these assumptions gives a lower bound without the assumptions.

**The process $\mathcal{P}_1$.** The process $\mathcal{P}_1$ maintains an $n \times d$ table $\Gamma$. A graph in $\mathcal{G}_1$ corresponds to a perfect matching between the table entries. That is, if there is an edge $(v, u)$ in the graph, and the edge is labeled $i$ from $v$'s side and $i'$ from $u$'s side, then $\Gamma(v, i) = (u, i')$ and $\Gamma(u, i') = (v, i)$. Thus, a random selection of a graph in $\mathcal{G}_1$ corresponds to a random selection of a perfect matching between the entries of $\Gamma$. Such a matching can be constructed iteratively, where in each iteration an unmatched entry in the table is selected *arbitrarily* and matched to a *uniformly selected* entry that is not yet matched. The process $\mathcal{P}_1$ fills the entries of $\Gamma$ in the course of answering the queries of the algorithm $\mathcal{A}$: Given a query $q_{t+1} = (v, i)$, the process $\mathcal{P}_1$ answers with a uniformly selected unmatched entry, $(u, i')$

**The process $\mathcal{P}_2$.** The process $\mathcal{P}_2$ maintains two tables: one $(n - c) \times d$ table, $\Gamma_{V \setminus S}$, and one $c \times d'$ table, $\Gamma_S$. The rows of $\Gamma_{V \setminus S}$ correspond to vertices in $V \setminus S$, and the rows of $\Gamma_S$ correspond to vertices in $S$. A random graph in $\mathcal{G}_2$ can be determined in the following iterative manner. In each step, a pair $(v, i)$ is selected arbitrarily among all pairs such that

- either there is already a row labeled by $v$ in one of the two tables, but the entry $(v, i)$ is not yet matched, or
- there is no row labeled by $v$.

In the latter case, we first select, uniformly at random, a row that is not yet labeled in one of the two tables, and label it by $v$. We then select, uniformly at random, an entry in one of the two tables that is not yet matched. If the row of the selected entry is not yet labeled, then we give it a random label (among all the labels in $\{1, \ldots, n\}$ that have not been used yet).

The process $\mathcal{P}_2$ fills the entries of $\Gamma_{V \setminus S}$ and $\Gamma_S$ in the course of answering the queries of the algorithm $\mathcal{A}$ in the following manner. First note that once a vertex $v$ that appears in either a query or an answer is determined to belong to $S$, then we may assume that $\mathcal{A}$ terminates, since it has evidence to distinguish between the two families (recall that the degree of a vertex is revealed when it appears in a query or an answer). Now, given a query $q_{t+1} = (v, i)$, if $v$ is a vertex that was not yet observed in the query-answer history (that is, it does not label any row), then $\mathcal{P}_2$ first determines whether $v$ belongs to $S$ or to $V \setminus S$, that is, if $v$ labels a row in $\Gamma_S$ or in $\Gamma_{V \setminus S}$. Let the number of vertices already determined to be in $V \setminus S$ be denoted by $b$ (so that $b \leq 2t$). With probability $\frac{c}{n-b}$, the vertex $v$ is determined to belong to $S$ (at which point $\mathcal{A}$ can terminate) and with probability $1 - \frac{c}{n-b}$, it is determined to belong to $V \setminus S$, so that it labels an unlabeled row in $\Gamma_{V \setminus S}$. Next, an entry that is not yet matched is selected uniformly among all such entries in $\Gamma_{V \setminus S}$ and $\Gamma_S$. If the selected entry is in $\Gamma_S$, then $\mathcal{A}$ can terminate. Otherwise, let $i'$ be the column to which the entry belongs (in $\Gamma_{V \setminus S}$). If the entry belongs to a row that is already labeled by some $u \in \{1, \ldots, n\}$, then $\mathcal{P}_2$ answers $(u, i')$, and if the row is unlabeled, then $\mathcal{P}_2$ uniformly selects a label $u \in \{1, \ldots, n\}$ among all row labels that are not yet used, and answers $(u, i')$.

**Analyzing the distributions on query-answer histories.** Consider $\mathcal{P}_2$, and observe that if the number of queries performed is $o(\sqrt{n})$, then the probability that a vertex $v$ in a query $(v, i)$ is determined to belong to $S$ is $o(\sqrt{n}) \cdot \frac{c}{n - o(\sqrt{n})} = o(\frac{1}{\sqrt{n}})$. The second observation about $\mathcal{P}_2$, the probability that the answer to a query $(v, i)$ will

be $(u, i')$, where $u \in S$ and $t$ queries have already been performed, equals the number of entries in $\Gamma_S$, which is $c \cdot d'$, divided by the number of entries in both tables that are not yet matched, which is $(n-c) \cdot d + c \cdot d' - 2t$. That is, we get $\frac{c \cdot d'}{(n-c) \cdot d + c \cdot d' - 2t} = O(\frac{1}{\sqrt{n}})$, and so the probability that such an event occurs in a sequence of $o(\sqrt{n})$ queries is $o(1)$.

Finally, for both processes, the probability that an answer to a query $q_{t+1} = (v, i)$ is $(u, i')$ for $u$ that has already appeared in the query-answer history is upper-bounded by $\frac{2t}{n} = o(\frac{1}{\sqrt{n}})$, and so the probability that such an event occurs in a sequence of $o(\sqrt{n})$ queries is $o(1)$. Therefore, in both processes, if the number of queries performed is $o(\sqrt{n})$, then for any algorithm $\mathcal{A}$, with probability $1 - o(1)$, the sequence of answers to the queries of $\mathcal{A}$ is a sequence of uniformly selected distinct pairs $(u, i')$. This implies that the statistical distance between the corresponding distributions on query-answer histories is $o(1)$, and so it is not possible to distinguish between a random graph in $\mathcal{G}_1$ and a random graph in $\mathcal{G}_2$ with probability greater than $\frac{1}{2} + o(1)$.

**The issue of multiple edges.** As defined above, the graphs may have multiple edges and self-loops. In order to avoid multiple edges and self-loops, the distribution on answers to queries given any particular history should be conditioned on the randomly constructed graph not containing any multiple edges and self-loops. While the precise form of the probability distribution on answers may be more complicated due to this conditioning, we only need to upper bound the probability of certain events. We first observe that we can use the same bound as above for the probability that a vertex $v$ in a query $(v, i)$ is determined to belong to $S$, and conclude that the probability that such an event occurs in a sequence of $o(\sqrt{n})$ queries is $o(1)$.

We next bound the probability that an answer to a query $q_{t+1} = (v, i)$ is $(u, i')$ for $u$ that has already appeared in the query-answer history. Our analysis follows a similar analysis in [BKKR10]. Starting with the family $\mathcal{G}_1$, consider the set of all graphs (with no multiple edges and no self-loops) that are consistent with the query-answer history. That is, they contain the subgraph $H$ corresponding to this history. Let $u$ be a vertex that appears in the query-answer history, and let $w$ be a vertex that does not appear in the history. Thus, the degree of $u$ in $H$ is at least 1 and the degree of $w$ in $H$ is 0. Let $\mathcal{C}_u$ denote the set of graphs in $\mathcal{G}_1$ that contain $H$ as a subgraph and in which there is an edge between $v$ and $u$, and let $\mathcal{C}_w$ denote the set of graphs in $\mathcal{G}_1$ that contain $H$ as a subgraph and in which there is an edge between $v$ and $w$. We claim that $|\mathcal{C}_w| \geq |\mathcal{C}_u|$, from which it follows that the probability that the answer to a neighbor query from $v$ is answered by any specific vertex $u$ that appears in the query-answer history is upper-bounded by the probability that it is answered by any specific vertex $w$ that has not yet appeared in the history.

To verify this we define an auxiliary bipartite graph in which there is a node on the left-hand side for every graph in $\mathcal{C}'_u = \mathcal{C}_u \setminus \mathcal{C}_w$ and a node on the right-hand side for every graph in $\mathcal{C}'_w = \mathcal{C}_w \setminus \mathcal{C}_u$. We put an edge in this bipartite graph between a node corresponding to graph $F \in \mathcal{C}'_u$ and a node corresponding to a graph $\tilde{F} \in \mathcal{C}'_w$ if the following holds. In $F$ (which contains the edge $(v, u)$ but not the edge $(v, w)$), there is vertex $x$ such that the edge $(x, w)$ belongs to $F$ and the edge $(x, u)$ does not belong to $F$, while in $\tilde{F}$ (which contains the edge $(v, w)$ but not the edge $(v, u)$), there is an edge between $u$ and $x$ but not between $w$ and $x$. The two graphs agree on all other edges. We next partition the nodes (graphs) on both sides of the auxiliary bipartite graph according to the size of the intersection of the neighborhood sets of $u$ and $w$, and note that there are edges in the auxiliary biparatite graph only between nodes that correspond to graphs for which this number is the same. Focusing on each such subbipartite auxiliary graph, the main

observation is that because $u$ is incident to at least one edge in $H$ (on which graphs in $\mathcal{C}'_u$ and $\mathcal{C}'_w$ cannot differ), while there is no such constraint on $w$, the degree of nodes on the left-hand side is upper-bounded by the degree of nodes on the right-hand side, implying that $|\mathcal{C}'_w| \geq |\mathcal{C}'_u|$, and hence $|\mathcal{C}_w| \geq |\mathcal{C}_u|$, as claimed.

The argument for graphs in $\mathcal{G}_2$ is essentially the same, where we consider the case that $H$ does not contains any vertex in $S$ (the probability that such an event occurs, allowing the algorithm to terminate, is addressed subsequently). In fact, since the degree of vertices in $S$ (which do not appear in the query-answer history) is larger than the degree of vertices in $V \setminus S$, the claim is even slightly stronger. Thus, it still holds that the probability that a neighbor query is answered by a vertex that has appeared in the query-answer history in a sequence of $o(\sqrt{n})$ queries is $o(1)$.

Finally consider the probability (for the case of $\mathcal{G}_2$) of answering a query $q_t = (v, i)$ with $(u, i')$, where $u \in S$. By the preceding analysis, there is a positive bias for answering with a vertex that has not appeared in the query-answer history (where this is the case for all vertices in $S$, or else the algorithm could have terminated). As observed previously, the number of vertices that have appeared in the query-answer history after $t$ queries is at most $2t$, and so the number of vertices in $V \setminus S$ that have not appeared in the query-answer history after $t$ queries is at least $n - c - 2t = \Omega(n)$. We claim that for any fixed choice of $u \in V \setminus S$ and $w \in S$ that have not appeared in the query-answer history, the probability that the query $q_t = (v, i)$ is answered with $(w, i')$ is $O(d'/d)$ times larger than the probability that it is answered with $(u, i'')$. This follows by an argument very similar to the one just presented for comparing between the probability of answering with a vertex that has appeared in the query-answer history, and the probability of answering with a vertex that has not appeared in this history.

Specifically, for $u \in V \setminus S$ and $w \in S$, we define $\mathcal{C}'_u$ and $\mathcal{C}'_w$ in the same manner as defined previously, and we define the auxiliary bipartite graph in the same manner. Here too we partition the auxiliary graph into subgraphs (with no edges between them) according to the size of the intersection of the set of neighbors of $u$ and the set of neighbors of $w$. As long as this size is at most $d/2$, the degree of nodes on the left-hand side is a factor of $O(d'/d)$-larger than the degree of nodes on the right-hand side. However, the relative number of graphs for which the size of this intersection is greater than $d/2$ is very small. It follows that the probability that a query $q_t = (v, i)$ is answered with $(u, i')$, where $u \in S$ in a sequence of $o(\sqrt{n})$ queries, is $o(1)$.

**4.3. Proof of item 3 in Theorem 3.** Similarly to the proof of item 2 in Theorem 3, to establish item 3 in Theorem 3 we show that for every $n$, every constant $c$, and every $\ell = \Omega(n^2)$, $\ell < n^3/(16c^2)$, there exist two families of $n$-vertex graphs for which the following holds. In both families the number of length-2 paths is $\Theta(\ell)$, but in one family this number is a factor $c$ larger than in the other family. However, it is not possible to distinguish with high constant probability between a graph selected randomly in one family and a graph selected randomly in the other family using $o(\frac{n^{3/2}}{\ell^{1/2}})$ queries. (Note that when $\ell = \Omega(n^3)$, and in particular, $\ell \geq n^3/(16c^2)$, the lower bound is $\Omega(1)$, which is trivial.)

The first family, $\mathcal{G}_1$, is identical to the one defined in the proof of item 2 in Theorem 3. That is, each graph is determined by $d = \lfloor \sqrt{2\ell/n} \rfloor$ matchings so that each vertex has degree $d$ and $\ell(G) = n \cdot \binom{d}{2} < \ell$. In the second family, denoted $\mathcal{G}_2$, each graph is defined as follows. There is a subset, $S$, of $s = \lceil \frac{4c\ell}{n^2} \rceil$ vertices, and a complete bipartite graph between $S$ and $V \setminus S$. In addition, there are $d - s$ perfect matchings between vertices in $V \setminus S$. For an illustration, see Figure 4.2. Thus, each vertex in $V \setminus S$ has degree $d$,

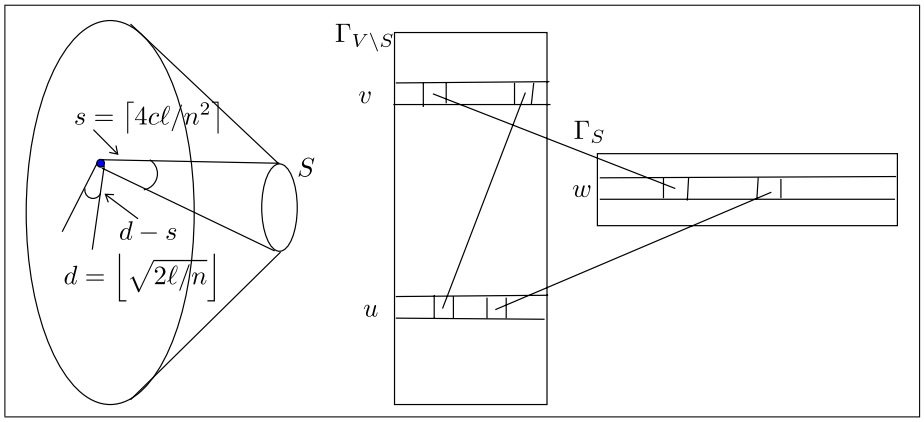


FIG. 4.2. *An illustration for the proof of item* 3 *in Theorem* 3. *On the left-hand side is a graph in* $\mathcal{G}_2$, *and on the right-hand side are the corresponding tables,* $\Gamma_{V\setminus S}$ *and* $\Gamma_S$. *A connecting line between a pair of entries indicates that there is an edge between the two corresponding vertices.*

just like in $\mathcal{G}_1$. Now, for every $G \in \mathcal{G}_2$, using our assumption that $\ell < n^3/(16c^2)$ so that $s < n/4$,

$$\ell(G) \geq s \cdot \binom{n-s}{2} > s \cdot \binom{3n/4}{2} = \left\lceil \frac{4c\ell}{n^2} \right\rceil \cdot \binom{3n/4}{2} > c\ell.$$

The argument for proving that no algorithm can distinguish with high constant probability between a graph selected randomly in $\mathcal{G}_1$ and a graph selected randomly in $\mathcal{G}_2$ is similar to the one presented in the proof of item 2 in Theorem 3, and is actually somewhat simpler. As in the proof of item 2 in Theorem 3, we define two processes, $\mathcal{P}_1$ and $\mathcal{P}_2$, where $\mathcal{P}_1$ is exactly as defined in the proof of Item 2 in Theorem 3.

**The process $\mathcal{P}_2$.** The process $\mathcal{P}_2$ maintains an $(n-s) \times d$ table, $\Gamma_{V\setminus S}$, and an $s \times (n-s)$ table, $\Gamma_S$. The rows of $\Gamma_{V\setminus S}$ correspond to vertices in $V \setminus S$, and the rows in $\Gamma_S$ correspond to vertices in $S$. A graph in $\mathcal{G}_2$ is determined by a perfect matching between the union of the entries in the two tables, where each row in $\Gamma_{V\setminus S}(v)$ contains exactly $s$ entries that are matched with entries of $\Gamma_S$, one from each row. Here too we may assume that once a vertex $v$ that appears in either a query or an answer is determined to belong to $S$ (i.e., to label a row in $\Gamma_S$), then $\mathcal{A}$ terminates, since it has evidence to distinguish between the two families.

Given a query $q_{t+1} = (v, i)$, if $v$ is a vertex that was not yet observed in the query-answer history, then $\mathcal{P}_2$ first determines whether it belongs to $S$ or to $V \setminus S$. Let the number of vertices already determined to be in $V \setminus S$ be denoted by $b$ (so that $b \leq 2t$). With probability $\frac{s}{n-b}$, the vertex $v$ is determined to belong to $S$ (at which point $\mathcal{A}$ can terminate) and with probability $1 - \frac{s}{n-b}$, it is determined to belong to $V \setminus S$, so that it labels a randomly chosen unlabeled row in $\Gamma_{V\setminus S}$. Next, the process decides whether the entry $(v, i)$ corresponds to an edge whose other endpoint is in $S$ or in $V \setminus S$. Let $b(v)$ be the number of entries in the row of $v$ that have already been determined. Then, with probability $\frac{s}{d-b(v)}$, the entry is matched to a uniformly selected entry in $\Gamma_S$ (so that $\mathcal{A}$ can terminate), and with probability $1 - \frac{s}{d-b(v)}$, it is matched to an entry in $\Gamma_{V\setminus S}$ that is not yet matched. This entry is selected as follows. For each row $r$ in $\Gamma_{V\setminus S}$ (labeled or unlabeled), let $b(r)$ be the number of entries in $r$ that are already matched.

Then a row $r$ is selected with probability $\frac{d-b(r)-s}{\sum_r (d-b(r)-s)}$. If the row is not yet labeled, then $\mathcal{P}_2$ uniformly selects a label $u \in \{1, \ldots, n\}$ among all unused row labels. The index $i'$ is selected uniformly among all entries in the row $r$ that are not yet matched.

**Analyzing the distributions on query-answer histories.** Consider $\mathcal{P}_2$, and observe that if the number of queries performed is $o(n^{3/2}/\ell^{1/2})$, then the probability that a vertex $v$ in a query $(v, i)$ is determined to belong to $S$ is

$$o(n^{3/2}/\ell^{1/2}) \cdot \frac{s}{n - o(n^{3/2}/\ell^{1/2})} = o(n^{3/2}/\ell^{1/2}) \cdot \frac{\lceil (4c\ell)/n^2 \rceil}{n} = o\left(\frac{\ell^{1/2}}{n^{3/2}}\right) = o(1).$$

The second observation about $\mathcal{P}_2$ is that for every $t = o(n^{3/2}/\ell^{1/2})$, the probability that the answer to a query $(v, i)$ will be $(u, i')$, where $u \in S$ is upper-bounded by

$$\frac{s}{d - b(v)} = \frac{\lceil (4c\ell)/n^2 \rceil}{\lfloor \sqrt{2\ell/n} \rfloor - o(n^{3/2}/\ell^{1/2})} = O(\ell^{1/2}/n^{3/2}),$$

and so the probability that such an event occurs in a sequence of $o(n^{3/2}/\ell^{1/2})$ queries is $o(1)$. Finally, for both processes, the probability that an answer to a query $q_{t+1} = (v, i)$ is $(u, i')$ for $u$ that has already appeared in the query-answer history is upper-bounded by $\frac{2t}{n} = o(n^{1/2}\ell^{1/2})$, and so the probability that such an event occurs in a sequence of $o(n^{3/2}/\ell^{1/2})$ queries is $o(n^2/\ell) = o(1)$.

Therefore, in both processes, if the number of queries performed is $o(n^{3/2}/\ell^{1/2})$, then for any algorithm $\mathcal{A}$, with probability $1 - o(1)$, the sequence of answers to the queries of $\mathcal{A}$ is a sequence of uniformly selected distinct pairs $(u, i')$. This implies that the statistical distance between the corresponding distributions on query-answer histories is $o(1)$, and so it is not possible to distinguish random graphs from the two families with probability greater than $\frac{1}{2} + o(1)$. The issue of multiple edges is dealt with as in the proof of item 2 in Theorem 3.

**5. Extending the algorithm to stars.** In this section we explain how our result for approximating the number of length-2 paths can be extended to larger stars. The new notations introduced in this section are collected in Table 5.1.

TABLE 5.1
*New notations for stars, their meaning, and the location of their exact definition, if appropriate.*

| Notation | Meaning | Exact definition |
|---|---|---|
| $\nu_s(G)$ | Number of $s$-stars in $G$ | |
| $\tilde{\nu}_s$ | Given estimate (const. factor) of $\nu_s(G)$ | |
| $\beta$ | $\epsilon/32s$ | |
| $\theta_1$ | Threshold parameter for Algorithm 3 | Step 1 in Algorithm 3 |
| $L$ | Set of indices of large buckets | Step 4 in Algorithm 3 |
| $\theta_2(p)$ | Threshold parameters for variant of Algorithm 2 | Equation (5.1) |
| $\nu_s^{(\sigma)}(G, \bar{L})$ | Certain numbers of stars | Definition 3 |
| $SIG$ | Indices of significant buckets | Definition 4 |

Recall that an *s-star* is a graph over $s+1$ vertices in which one single vertex (the star *center*) is adjacent to all other vertices (and there are no edges between the other vertices). In particular, a length-2 path is a 2-star. The algorithm for approximating the number of $s$-stars for $s > 2$ is a natural extension of the algorithm we presented for the case of $s = 2$, and its analysis is very similar. Here we describe the modifications in the algorithm and its analysis. Recall that $\nu_s(G)$ denotes the number of $s$-stars in a graph $G$. Here too we assume the algorithm is given a rough estimate $\tilde{\nu}_s$ for $\nu_s(G)$ such that $\frac{1}{2}\nu_s(G) \leq \tilde{\nu}_s \leq 2\nu_s(G)$, and this assumption is removed in the same manner as in the case of length-2 paths. We assume for simplicity that $s$ is a constant, though it can also be a very slowly growing function of $n$ (since the dependence on $s$ is exponential).

The variant of Algorithm 2 (referred to in Algorithm 3) used to get the estimates $\{\hat{e}_{i,j}\}_{j \notin L}$ is the following. For each $0 \leq p \leq i$, let

$$\theta_2(p) \stackrel{\text{def}}{=} \frac{\epsilon^{\frac{s+1}{s}} \tilde{\nu}_s^{\frac{1}{s}}}{c_2(s) t^{\frac{2s+1}{s}} (1+\beta)^{\frac{p}{s}}}, \tag{5.1}$$

where $c_2(s)$ grows at most exponentially with $s$. The minimum value $p_0$ of $p$ is still the smallest value of $p$ satisfying $\frac{1}{4}\theta_2(p+1) \leq n$. The sample size $s^{(p)}$ is still

$$s^{(p)} = \Theta\left(\frac{n}{\theta_2(p)} \left(\frac{t}{\beta}\right)^2 \log\, t\right), \tag{5.2}$$

and in step 3c we have the following:

3c. If $|S_i^{(p)}| < \frac{s^{(p)}}{n} \cdot \frac{1}{4(1+\beta)}\theta_2(p)$, then go to step 4. Else, if $|S_i^{(p)}| > \frac{s^{(p)}}{n} \cdot \frac{4\tilde{\nu}_s}{\binom{(1+\beta)^{i-1}}{s}}$, then terminate and return 0.

ALGORITHM 3. (Estimating the number of $s$-stars in $G = (V, E)$)
*Input:* $\epsilon$, $s$, *and* $\tilde{\nu}_s$.

1. *Let* $\beta \stackrel{\text{def}}{=} \frac{\epsilon}{32s}$, $t \stackrel{\text{def}}{=} \left\lceil \log_{(1+\beta)} n \right\rceil$, *and*

$$\theta_1 \stackrel{\text{def}}{=} \frac{\epsilon^{\frac{s}{s+1}} \tilde{\nu}_s^{\frac{1}{s+1}}}{c_1(s) t^{\frac{2s}{s+1}}},$$

   *where* $c_1(s)$ *will be set in the analysis.*
2. *Uniformly and independently select* $\Theta\left(\frac{n}{\theta_1} \cdot \frac{\log t}{\epsilon^2}\right)$ *vertices from* $V$, *and let* $S$ *denote the multiset of selected vertices (that is, we allow repetitions).*
3. *For* $i = 0, \ldots, t$, *determine* $S_i = S \cap B_i$ *by performing a degree query on every vertex in* $S$.
4. *Let* $L = \left\{i \,:\, \frac{|S_i|}{|S|} \geq 2\frac{\theta_1}{n}\right\}$. *If* $\max_{i \in L}\left\{\binom{(1+\beta)^{i-1}}{s} \cdot \theta_1\right\} > 4\tilde{\nu}_s$, *then terminate and return 0.*
5. *For each* $i \in L$ *run a slight variant of Algorithm 2 (that is described below) to get estimates* $\{\hat{e}_{i,j}\}_{j \notin L}$ *for* $\{|E_{i,j}|\}_{j \notin L}$.
6. *Output*

$$\hat{\nu}_s = \sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{s} + \sum_{j \notin L} \frac{1}{s} \sum_{i \in L} \hat{e}_{i,j} \binom{(1+\beta)^j - 1}{s-1}.$$

THEOREM 4. *If* $\frac{1}{2}\nu_s(G) \le \tilde{\nu}_s \le 2\nu_s(G)$*, then with probability at least* $2/3$*, the output,* $\hat{\nu}_s$*, of Algorithm* 3 *satisfies* $\hat{\nu}_s = (1 \pm \epsilon) \cdot \nu_s(G)$*. The query complexity and running time of the algorithm are*

$$O\left(\frac{n}{\tilde{\nu}_s^{\frac{1}{s+1}}} + \min\left\{n^{1-\frac{1}{s}}, \frac{n^{s-\frac{1}{s}}}{\tilde{\nu}_s^{1-\frac{1}{s}}}\right\}\right) \cdot \text{poly}(\log\, n, 1/\epsilon).$$

As noted previously, the assumption that the algorithm has an estimate $\tilde{\nu}_s$ for $\nu_s(G)$ is removed similarly to the way it was removed in the case of 2-stars.

**5.1. Analyzing the complexity of Algorithm 3.** As in the case of Algorithm 1, the running time of Algorithm 3 is linear in its query complexity, and hence it suffices to bound the latter. Recall that we restrict our attention to constant $s$, and hence, in our bounds, we shall ignore terms that depend only on $s$. The query complexity of Algorithm 3 is the sum of $\Theta(\frac{n}{\theta_1} \cdot \frac{\log t}{\epsilon^2}) = O(\frac{n}{\tilde{\nu}_s^{\frac{1}{s+1}}}) \cdot \text{poly}(\log\, n, 1/\epsilon)$ (the size of the sample selected in step 2 of the algorithm) and the number of queries performed in the executions of the variant of Algorithm 2. In order to bound the latter, we first observe that if Algorithm 3 did not terminate in step 4, then

$$\forall i \in L\colon (1+\beta)^i = O\left(\frac{\tilde{\nu}_s^{\frac{1}{s+1}} \cdot t^{\frac{2}{s+1}}}{\epsilon^{\frac{1}{s+1}}}\right). \tag{5.3}$$

Similarly, if the variant of Algorithm 2 did not terminate in any of its executions in step 3c (where this step is as described following (5.2)), then

$$\forall i \in L \quad \text{and} \quad p_0 \le p \le i\colon |S_i^{(p)}| = O\left(\frac{s^{(p)}}{n} \cdot \frac{\tilde{\nu}_s}{\binom{(1+\beta)^{i-1}}{s}}\right). \tag{5.4}$$

In addition, it trivially always holds that $|S_i^{(p)}| \le s^{(p)} = O(\frac{n}{\theta_2(p)}(\frac{t}{\beta})^2 \log\, t)$. Recall that $p$ runs from $i$ down to $p_0$, where $p_0$ is the smallest value of $p$ satisfying $\frac{1}{4}\theta_2(p+1) \le n$, where $\theta_2(p)$ is as defined in (5.1). That is,

$$p_0 = \left\lfloor \log_{1+\beta} \frac{\epsilon^{s+1}\tilde{v}_s}{c_2'(s) \cdot t^{2s+1} n^s} \right\rfloor$$

for an appropriate choice of $c_2'(s)$. This implies that if

$$\tilde{\nu}_s \le \frac{c_2'(s) \cdot t^{2s+1}}{\epsilon^{s+1}} \cdot n^s,$$

then $p_0 = 0$, and otherwise it may be larger. Therefore, the total number of queries performed in the executions of the variant of Algorithm 2 is upper-bounded by

$$\sum_{i\in L}\sum_{p=p_0}^{i}\left(s^{(p)}+\min\left\{s^{(p)},\frac{s^{(p)}}{n}\cdot\frac{4\tilde{\nu}_s}{\binom{(1+\beta)^{i-1}}{s}}\right\}\cdot g^{(p)}\right)$$

$$\leq\sum_{i\in L}i\cdot s^{(i)}+\sum_{i\in L}\sum_{p=p_0}^{i}\min\left\{s^{(p)},\frac{s^{(p)}}{n}\cdot\frac{4\tilde{\nu}_s}{\binom{(1+\beta)^{i-1}}{s}}\right\}\cdot g^{(p)}. \tag{5.5}$$

For the first summand in (5.5) we apply (5.3) and get (similarly to (3.6))

$$\begin{aligned}\sum_{i\in L}i\cdot s^{(i)}&\leq\sum_{i\in L}t\cdot O\left(\frac{n\cdot t^{4+\frac{1}{s}}\log\,t\cdot(1+\beta)^{\frac{i}{s}}}{\epsilon^{3+\frac{1}{s}}\cdot\tilde{\nu}_s^{\frac{1}{s}}}\right)\\&=O\left(\frac{n\cdot t^{6+\frac{1}{s}}\log\,t}{\epsilon^{3+\frac{1}{s}}\cdot\tilde{\nu}_s^{\frac{1}{s}}}\cdot\left(\frac{\tilde{\nu}_s^{\frac{1}{s+1}}\cdot t^{3+\frac{1}{s}}}{\epsilon^{\frac{1}{s+1}}}\right)^{\frac{1}{s}}\right)\\&=O\left(\frac{n}{\tilde{\nu}_s^{\frac{1}{s+1}}}\right)\cdot\mathrm{poly}(\log\,n,1/\epsilon).\end{aligned} \tag{5.6}$$

Turning to the second summand in (5.5), the following is derived similarly to (3.7):

$$\begin{aligned}&\sum_{i\in L}\sum_{p=p_0}^{i}\min\left\{s^{(p)},\frac{s^{(p)}}{n}\cdot\frac{4\tilde{\nu}_s}{\binom{(1+\beta)^{i-1}}{s}}\right\}\cdot g^{(p)}\\&=\sum_{i\in L}\sum_{p=p_0}^{i}O\left(\min\left\{\frac{n\cdot(1+\beta)^{\frac{p}{s}}}{\tilde{\nu}_s^{\frac{1}{s}}},\frac{\tilde{\nu}_s^{1-\frac{1}{s}}}{(1+\beta)^{si-\frac{p}{s}}}\right\}\cdot(1+\beta)^{i-p}\right)\cdot\mathrm{poly}(\log\,n,1/\epsilon)\\&=\sum_{i\in L}O\left(\min\left\{\frac{n\cdot(1+\beta)^{i}}{\tilde{\nu}_s^{\frac{1}{s}}},\frac{\tilde{\nu}_s^{1-\frac{1}{s}}}{(1+\beta)^{(s-1)i}}\right\}\cdot\mathrm{poly}(\log\,n,1/\epsilon)\right)\\&\quad\cdot(1+\beta)^{-(1-\frac{1}{s})p_0}\cdot\sum_{k=0}^{i-p_0}(1+\beta)^{-(1-\frac{1}{s})k}.\end{aligned} \tag{5.7}$$

In order to bound the expression in (5.7), we first note that if $(1+\beta)^i\leq\frac{\tilde{\nu}_s^{1/s}}{n^{1/s}}$, then $\frac{n\cdot(1+\beta)^i}{\tilde{\nu}_s^{1/s}}\leq n^{1-1/s}$, while if $(1+\beta)^i\geq\frac{\tilde{\nu}_s{}^{1/s}}{n^{1/s}}$, then $\frac{\tilde{\nu}_s{}^{1-1/s}}{(1+\beta)^{(s-1)i}}\leq n^{1-1/s}$ as well. Since $(1+\beta)^{-(1-1/s)p_0}=1$ and $\sum_{k=0}^{i-p_0}(1+\beta)^{-(1-1/s)k}=O(1/\beta)$, if $p_0=0$, then the right-hand side of (5.7) is upper-bounded by

$$O(n^{1-\frac{1}{s}})\cdot\mathrm{poly}(\log\,n,1/\epsilon). \tag{5.8}$$

If $p_0>0$, then the bound in (5.8) should be multiplied by $(1+\beta)^{-(1-1/s)p_0}$. By the definition of $p_0$, we have that $(1+\beta)^{-(1-1/s)p_0}=O(\frac{n^{s-1}}{\tilde{\nu}_s^{1-1/s}})\cdot\mathrm{poly}(\log\,n,1/\epsilon)$, and so we get the (tighter) bound:

$$O(n^{1-\frac{1}{s}} \cdot (1+\beta)^{-(1-\frac{1}{s})p_0}) \cdot \text{poly}(\log\, n, 1/\epsilon) = O\left(\frac{n^{s-\frac{1}{s}}}{\tilde{\nu}_s^{1-\frac{1}{s}}}\right) \cdot \text{poly}(\log\, n, 1/\epsilon). \tag{5.9}$$

The total number of queries performed in the executions of the variant of Algorithm 2 is hence upper-bounded by

$$O\left(\frac{n}{\tilde{\nu}_s^{\frac{1}{s+1}}} + \min\left\{n^{1-\frac{1}{s}}, \frac{n^{s-\frac{1}{s}}}{\tilde{\nu}_s^{1-\frac{1}{s}}}\right\}\right) \cdot \text{poly}(\log\, n, 1/\epsilon). \tag{5.10}$$

**5.2. Analyzing the correctness of Algorithm 3.** We first note that the size of the sample $S$ selected by Algorithm 3 is $\Theta(\frac{n}{\theta_1} \cdot \frac{\log t}{\epsilon^2})$, that is, the same, as a function of $\theta_1$, as the sample size selected by Algorithm 1. Therefore, Lemma 1 and Corollary 2 hold as is (for $\theta_1$ as defined in step 1 of Algorithm 3). The first part of Corollary 2 implies that (with high constant probability) the estimate $\sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{s}$ is close to the actual number of $s$-stars whose center belongs to a bucket $B_i$ such that $i \in L$. It also implies (similar to what was shown in the case of 2-stars) that Algorithm 3 does not terminate in step 4 (with high constant probability).

The remainder of the analysis deals with the quality of the estimate for the number of $s$-stars in $G$ whose center is not in $L$. First observe that by the second part of Corollary 2 we have that with high constant probability,

$$\sum_{j \notin L} |B_j| < 4\theta_1 t. \tag{5.11}$$

In the next definition we generalize Definition 1.

DEFINITION 3. *For $j \notin L$ and $\sigma \in \{1, 2, \ldots, s+1\}$, let $\nu_s^{(\sigma)}(j, G, \bar{L})$ denote the number of $s$-stars in $G$ whose center belongs to $B_j$ and such that the number of vertices in the star that belong to $B_k$ for $k \notin L$ (including $j$) is $\sigma$. Let $\nu_s^{(\sigma)}(G, \bar{L}) = \sum_{j \notin L} \nu_s^{(\sigma)}(j, G, \bar{L})$ and let $\nu_s(j, G, \bar{L}) = \sum_{\sigma=1}^{s+1} \nu_s^{(\sigma)}(j, G, \bar{L})$.*

We first observe that $\sum_{\sigma=2}^{s+1} \nu_s^{(\sigma)}(G, \bar{L})$ (stars that include at least one vertex from $B_j$ such that $j \notin L$ in addition to the center vertex) is relatively small (with high probability).

LEMMA 12. *With high constant probability, $\sum_{\sigma=2}^{s+1} \nu_s^{(\sigma)}(G, \bar{L}) \leq \frac{\epsilon}{8} \nu_s(G)$.*

*Proof.* We first observe that since the total number of $s$-stars is $\nu_s(G)$, for every bucket $B_j$ we have that $\binom{(1+\beta)^{j-1}+1}{s} \leq \nu_s(G)/|B_j|$. Hence,

$$(1+\beta)^j \leq (1+\beta) \cdot \left(\left(\frac{s! \cdot \nu_s(G)}{|B_j|}\right)^{\frac{1}{s}} + (s-1)\right).$$

For each $j \notin L$, we have that $\sum_{\sigma=2}^{s+1} \nu_s^{(\sigma)}(j, G, \bar{L})$ is the number of $s$-stars whose center, $v$, is in $B_j$ and that have at least one additional vertex in a bucket $B_k$, where $k \notin L$. (The remaining $s-1$ vertices may belong to any of the at most $(1+\beta)^j$ neighbors of $v$.) Therefore,

$$\sum_{\sigma=2}^{s+1} \nu_s^{(\sigma)}(G,\bar{L}) \leq \sum_{j\notin L}|B_j| \cdot \sum_{k\notin L}|B_k| \cdot \binom{(1+\beta)^j}{s-1}$$

$$\leq 4\theta_1 t \cdot \sum_{j\notin L}|B_j| \cdot \left(\binom{(1+\beta)\cdot\left(\left(\frac{s!\cdot\nu_s(G)}{|B_j|}\right)^{\frac{1}{s}}+(s-1)\right)}{s-1}\right). \tag{5.12}$$

Let $J_1 = \{j \notin L : (\frac{s!\cdot\nu_s(G)}{|B_j|})^{\frac{1}{s}} \leq s\}$ and let $J_2 = \{j \notin L : (\frac{s!\cdot\nu_s(G)}{|B_j|})^{\frac{1}{s}} > s\}$. Then,

$$4\theta_1 t \cdot \sum_{j\in J_1}|B_j| \cdot \left(\binom{(1+\beta)\cdot\left(\left(\frac{s!\cdot\nu_s(G)}{|B_j|}\right)^{\frac{1}{s}}+(s-1)\right)}{s-1}\right)$$

$$\leq 16(\theta_1\cdot t)^2 \cdot 2^{4s} < \frac{\epsilon}{32}\tilde{\nu}_s, \tag{5.13}$$

where the last inequality holds for $c_1(s) \geq 2^{4s+9}$. Turning to $J_2$, we have that

$$4\theta_1 t \cdot \sum_{j\in J_2}|B_j| \cdot \left(\binom{(1+\beta)\cdot\left(\left(\frac{s!\cdot\nu_s(G)}{|B_j|}\right)^{\frac{1}{s}}+(s-1)\right)}{s-1}\right)$$

$$\leq 4\theta_1 t \cdot \sum_{j\in J_2}|B_j| \cdot \binom{4\cdot\left(\frac{s!\cdot\nu_s(G)}{|B_j|}\right)^{\frac{1}{s}}}{s-1}$$

$$\leq 4\theta_1 t \cdot \sum_{j\in J_2}\frac{4^s(s!)^{\frac{1}{s}}}{(s-1)!}\cdot|B_j|^{\frac{1}{s}}\cdot(\nu_s(G))^{1-\frac{1}{s}}$$

$$\leq \frac{4^s(s!)^{\frac{1}{s}}}{(s-1)!}\cdot 4\theta_1 t\cdot t\cdot(4\theta_1)^{\frac{1}{s}}\cdot(\nu_s(G))^{1-\frac{1}{s}}$$

$$\leq \frac{4^{s+2}(s!)^{\frac{1}{s}}}{(s-1)!}\cdot t^2\cdot\theta_1^{1+\frac{1}{s}}\cdot(\nu_s(G))^{1-\frac{1}{s}}$$

$$\leq \frac{4^{s+2}(s!)^{\frac{1}{s}}}{(s-1)!}\cdot t^2\cdot\left(\frac{\epsilon^{\frac{s}{s+1}}\tilde{\nu}_s^{\frac{1}{s+1}}}{c_1(s)t^{\frac{2s}{s+1}}}\right)^{1+\frac{1}{s}}\cdot(\nu_s(G))^{1-\frac{1}{s}}$$

$$\leq \frac{\epsilon}{32}\tilde{\nu}_s, \tag{5.14}$$

where the last inequality holds for an appropriate choice of $c_1(s)$. The lemma follows by combining (5.12), (5.13), and (5.14). ∎

We next modify the notion of significant buckets (for buckets $B_j$ such that $j \notin L$).

Definition 4 (significant small buckets). *For every $j \notin L$ we say that $j$ is* significant *if*

$$|B_j| \cdot \binom{(1+\beta)^j}{s} \geq \frac{\epsilon}{c_3(s)t}\tilde{\nu}_s,$$

*where $c_3(s)$ grows at most exponentially with $s$. We denote the set of indices of significant buckets $B_j$ (where $j \notin L$) by* SIG.

Note that by the definition of SIG,

$$\sum_{j \notin L, j \notin SIG} \nu_s(j) < \frac{\epsilon}{c_3(s)} \tilde{\nu}_s \leq \frac{2\epsilon}{c_3(s)} \nu_s(G). \tag{5.15}$$

The next lemma is proved very similarly to Lemma 4.

LEMMA 13. *If $j \in SIG$, then for every $r$ such that $|B_{i,j,r}| > 0$ for some $i$, we have that*

$$|E_j| \geq \frac{c_4(s)t^2}{\epsilon} \cdot \theta_2(r) \cdot (1+\beta)^{r-1}$$

*for $c_4(s) = (c_2(s)/c_3(s)^{1/s}) \cdot (s!)^{1/s}$.*

*Proof.* Since $j$ is significant,

$$\binom{(1+\beta)^j}{s} \geq \frac{\epsilon}{c_3(s)t|B_j|} \tilde{\nu}_s.$$

Using the fact that

$$\binom{(1+\beta)^j}{s} \leq \frac{((1+\beta)^j)^s}{s!},$$

we get that

$$(1+\beta)^j > \left(\frac{s!}{c_3(s)}\right)^{\frac{1}{s}} \cdot \left(\frac{\epsilon \tilde{\nu}_s}{t|B_j|}\right)^{\frac{1}{s}}.$$

Since the graph contains no multiple edges, $|B_j| \geq (1+\beta)^r$ for every $r$ such that $B_{i,j,r}$ is not empty. Therefore (similar to the proof of Lemma 4),

$$\begin{aligned}
|E_j| &\geq |B_j| \cdot (1+\beta)^{j-1} \\
&\geq \frac{(s!)^{\frac{1}{s}}}{c_3(s)^{\frac{1}{s}}(1+\beta)} \cdot \frac{\epsilon^{\frac{1}{s}} \tilde{\nu}_s^{\frac{1}{s}} |B_j|^{1-\frac{1}{s}}}{t^{\frac{1}{s}}} \\
&\geq \frac{(s!)^{\frac{1}{s}}}{c_3(s)^{\frac{1}{s}}(1+\beta)t^{\frac{1}{s}}} \cdot \epsilon^{\frac{1}{s}} \tilde{\nu}_s^{\frac{1}{s}} (1+\beta)^{r(1-\frac{1}{s})} \\
&\geq \frac{(c_2(s)/c_3(s)^{\frac{1}{s}})t^2}{\epsilon} \theta_2(r) \cdot (1+\beta)^{r-1},
\end{aligned}$$

and the proof is completed. ∎

We now turn to explaining what needs to be modified in the analysis of the variant of Algorithm 2 that is used in Algorithm 3. Recall that $s^{(p)}$ denotes the size of the sample $S^{(p)}$, where $s^{(p)} = \Theta(\frac{n}{\theta_2(p)} \cdot (\frac{t}{\beta})^2 \log\, t)$. That is, the sample size is the same, as a function of $\theta_2(p)$, as in Algorithm 2. Hence, Lemma 5 and Corollary 6 hold as is, and here we also have that with high constant probability, the variant of Algorithm 2 does not terminate in step 3c. Lemmas 7 and 8 also hold without any changes. We do, however, need to modify (the second part of) Lemma 9, and the modified version is stated next.

LEMMA 14. *For an appropriate choice of $c_2(s)$ (in the definition of $\theta_2(\cdot)$, that is, (5.1)) and of $c_3(s)$ (in Definition 2), with high constant probability, for all $j \notin L$, if $j \in SIG$, then*

$$\left(1-\frac{\epsilon}{8}\right)\sum_{i\in L}|E_{i,j}|-\frac{\epsilon}{16}|E_j| \leq \sum_{i\in L}\hat{e}_{i,j} \leq \left(1+\frac{\epsilon}{4}\right)|E_j|,$$

*and if $j \notin SIG$, then*

$$\sum_{i\in L}\frac{1}{s}\,\hat{e}_{i,j}\cdot\binom{(1+\beta)^j}{s-1} \leq \frac{\epsilon}{4t}\,\nu_s(G).$$

The proof of Lemma 14 is very similar to the proof of Lemma 9. The only difference is that here we use the modified definition of significant buckets (Definition 4) and the corresponding lemma (Lemma 13) rather than the original definition (Definition 2) and lemma (Lemma 4). Note that in both lemmas, if $j \in SIG$, then $|E_j|$ is lower bounded by $\Omega(\frac{t^2}{\epsilon}\theta_2(r)\cdot(1+\beta)^r)$ (for each $r$ such that $B_{i,j,r}$ is not empty). As shown in the proof of Lemma 9 (based on the lemmas that hold as is for the case of $s$-stars), when $|B_{i,j,r}| \leq \frac{1}{4}\theta_2(r)$, the upper bound on $\hat{e}_{i,j,r}$ is of the order of $\theta_2(r)\cdot(1+\beta)^r$ (see (3.20)), and when $|B_{i,j,r}| > \frac{1}{4}\theta_2(r)$, the additive term in the deviation from $|E_{i,j,r}|$ is of the same order (see (3.31) and (3.32)). Therefore, when $j \in SIG$, they both translate to expressions of the form $\frac{\epsilon}{c_4(s)t^2}|E_j|$, as in the proof of Lemma 9 (see (3.21) and (3.33)). On the other hand, when $j \notin SIG$, we need to show that

$$\theta_2(r)\cdot(1+\beta)^r\cdot\binom{(1+\beta)^j}{s-1} \leq \frac{\epsilon}{c_5(s)t^3}\,\nu_s(G)$$

for an appropriate choice of $c_5(s)$.

If $j \notin SIG$, then $\binom{(1+\beta)^j}{s} \leq \frac{\epsilon}{c_3 t}\frac{\tilde{\nu}_s}{|B_j|}$, and so $(1+\beta)^j \leq (\frac{\epsilon s!}{c_3 t}\cdot\frac{\tilde{\nu}_s}{|B_j|})^{1/s}+(s-1)$. If $(\frac{\epsilon s!}{c_3 t}\frac{\tilde{\nu}_s}{|B_j|})^{1/s} \leq s$, then $(1+\beta)^j \leq 2s$ so that

$$\begin{aligned}
\theta_2(r)\cdot(1+\beta)^r\cdot\binom{(1+\beta)^j}{s-1} &\leq \frac{\epsilon^{\frac{s+1}{s}}\tilde{\nu}_s^{\frac{1}{s}}}{c_2(s)t^{2+\frac{1}{s}}(1+\beta)^{\frac{r}{s}}}\cdot(1+\beta)^r\cdot 2^{2s}\\
&\leq \frac{\epsilon^{\frac{s+1}{s}}\tilde{\nu}_s^{\frac{1}{s}}}{c_2(s)t^{2+\frac{1}{s}}}\cdot(1+\beta)^{(1-\frac{1}{s})}\cdot 2^{2s}\\
&\leq \frac{\epsilon^{\frac{s+1}{s}}\tilde{\nu}_s^{\frac{1}{s}}}{c_2(s)t^{2+\frac{1}{s}}}\cdot(4\theta_1)^{1-\frac{1}{s}}\cdot 2^{2s}\\
&\leq \frac{\epsilon^{\frac{s+1}{s}}\tilde{\nu}_s^{\frac{1}{s}}}{c_2(s)t^{2+\frac{1}{s}}}\cdot\left(\frac{4\epsilon^{\frac{s}{s+1}}\tilde{\nu}_s^{\frac{1}{s+1}}}{c_1(s)t^{\frac{2s}{s+1}}}\right)^{1-\frac{1}{s}}\cdot 2^{2s}\\
&= \frac{\epsilon}{c_5(s)t^3}\,\nu_s(G)
\end{aligned}$$

for an appropriate setting of $c_5(s)$ (that is a function of $c_1(s)$, $c_2(s)$, and $s$). We have used the fact that $(1+\beta)^r \leq |B_j|$ (since there are no multiple edges) and that $j \notin L$ (so that $|B_j| \leq 4\theta_1 = \frac{4\epsilon^{\frac{s}{s+1}}\tilde{\nu}_s^{\frac{1}{s+1}}}{c_1(s)t^{\frac{2s}{s+1}}}$). On the other hand, if $(\frac{\epsilon s!}{c_3(s)t}\frac{\tilde{\nu}_s}{|B_j|})^{1/s} > s$, then $(1+\beta)^j \leq 2(\frac{\epsilon s!}{c_3(s)t}\frac{\tilde{\nu}_s}{|B_j|})^{1/s}$ so that

$$\theta_2(r)\cdot(1+\beta)^r\cdot\binom{(1+\beta)^j}{s-1}\le\frac{\epsilon^{\frac{s+1}{s}}\tilde{\nu}_s^{\frac{1}{s}}}{c_2(s)t^{2+\frac{1}{s}}}\cdot(1+\beta)^{r(1-1/s)}\cdot 2^{s-1}\cdot\left(\frac{\epsilon s!}{c_3(s)t}\cdot\frac{\tilde{\nu}_s}{|B_j|}\right)^{1-\frac{1}{s}}$$
$$<\frac{s2^s}{c_2(s)c_3(s)}\cdot\frac{\epsilon}{t^3}\cdot\tilde{\nu}_s=\frac{\epsilon}{c_5'(s)t^3}\nu_s(G)$$

for an appropriate setting of $c_5'(s)$, which is a function of $c_2(s)$, $c_3(s)$, and $s$ (where here too we used the fact that $(1+\beta)^r\le|B_j|$).

The remainder of the proof is essentially as in the proof of Lemma 9, where the only difference is in the constraints on $c_2(s)$ and $c_3(s)$.

**5.3. Putting it all together: Proving the first part of Theorem 4.** Recall that

$$\hat{\nu_s}=\sum_{i\in L}n\cdot\frac{|S_i|}{|S|}\cdot\binom{(1+\beta)^i}{s}+\sum_{j\notin L}\frac{1}{s}\sum_{i\in L}\hat{e}_{i,j}\cdot\binom{(1+\beta)^j-1}{s-1}. \tag{5.16}$$

Let $\nu_s(G,L)$ denote the number of $s$-stars in $G$ whose center belongs to a bucket $B_i$ such that $i\in L$, and let $\nu_s(G,\bar{L})$ denote the number of $s$-stars whose center belongs to a bucket $B_j$ such that $j\notin L$ (so that $\nu_s(G,L)+\nu_s(G,\bar{L})=\nu_s(G)$). By the first part of Corollary 2 (and the setting of $\beta=\frac{\epsilon}{32s}$), we have that with high constant probability

$$\sum_{i\in L}n\cdot\frac{|S_i|}{|S|}\cdot\binom{(1+\beta)^i}{s}=\left(1\pm\frac{\epsilon}{4}\right)\nu_s(G,L). \tag{5.17}$$

Turning to the second summand in (5.16), by Lemma 14,

$$\begin{aligned}
&\sum_{j\notin L}\frac{1}{s}\sum_{i\in L}\hat{e}_{i,j}\cdot\binom{(1+\beta)^j-1}{s-1}\\
&\quad=\sum_{j\notin L,j\in SIG}\frac{1}{s}\sum_{i\in L}\hat{e}_{i,j}\cdot\binom{(1+\beta)^j-1}{s-1}\\
&\qquad+\sum_{j\notin L,j\notin SIG}\frac{1}{s}\sum_{i\in L}\hat{e}_{i,j}\cdot\binom{(1+\beta)^j-1}{s-1}\\
&\quad\le\sum_{j\notin L,j\in SIG}\frac{1}{s}\cdot\left(1+\frac{\epsilon}{4}\right)|E_j|\cdot\binom{(1+\beta)^j-1}{s-1}+\frac{\epsilon}{4}\nu_s(G)\\
&\quad\le\left(1+\frac{\epsilon}{4}\right)\cdot\sum_{j\notin L}\frac{1}{s}|E_j|\cdot\binom{(1+\beta)^j-1}{s-1}+\frac{\epsilon}{4}\nu_s(G)\\
&\quad\le\left(1+\frac{\epsilon}{2}\right)\nu_s(G,\bar{L})+\frac{\epsilon}{4}\nu_s(G). \qquad (5.18)
\end{aligned}$$

In the other direction, recall that $\nu_s^{(\sigma)}(G,\bar{L})=\sum_{j\notin L}\nu_s^{(\sigma)}(j,G,\bar{L})$, where for $j\notin L$ and $\sigma\in\{1,\ldots,s+1\}$, we let $\nu_s^{(\sigma)}(j,G,\bar{L})$ denote the number of $s$-stars whose center belongs to $B_j$ and such that the number of vertices in the star that belong to $B_k$ for $k\notin L$ (including $j$) is $\sigma$,

$$\begin{aligned}
&\sum_{j\notin L}\frac{1}{s}\sum_{i\in L}\hat{e}_{i,j}\cdot\binom{(1+\beta)^j-1}{s-1}\\
&\quad\geq\sum_{j\notin L,j\in SIG}\frac{1}{s}\left(\sum_{i\in L}\left(1-\frac{\epsilon}{8}\right)|E_{i,j}|-\frac{\epsilon}{16}|E_j|\right)\cdot\binom{(1+\beta)^j-1}{s-1}\\
&\quad\geq\sum_{j\notin L}\frac{1}{s}\sum_{i\in L}\left(1-\frac{\epsilon}{8}\right)|E_{i,j}|\cdot\binom{(1+\beta)^j-1}{s-1}-\frac{(1+\beta)^{s-1}\epsilon}{16}\nu_s(G) \qquad (5.19)\\
&\qquad-\sum_{j\notin L,j\notin SIG}\frac{1}{s}\sum_{i\in L}\left(1-\frac{\epsilon}{8}\right)|E_{i,j}|\cdot\binom{(1+\beta)^j-1}{s-1}\\
&\quad\geq\left(1-\frac{\epsilon}{8}\right)\cdot\sum_{\sigma=1}^{s}\frac{s-(\sigma-1)}{s}\nu_s^{(\sigma)}(G,\bar{L})-\frac{\epsilon}{4}\nu_s(G)
\end{aligned}$$

$$\begin{aligned}
&\geq\left(1-\frac{\epsilon}{8}\right)\cdot\sum_{\sigma=1}^{s+1}\nu_s^{(\sigma)}(G,\bar{L})-\sum_{\sigma=2}^{s+1}\nu_s^{(\sigma)}(G,\bar{L})-\frac{\epsilon}{4}\nu_s(G) \qquad (5.20)\\
&\geq\left(1-\frac{\epsilon}{8}\right)\nu_s(G,\bar{L})-\frac{\epsilon}{8}\nu_s(G)-\frac{\epsilon}{4}\nu_s(G)\\
&=\left(1-\frac{\epsilon}{8}\right)\nu_s(G,\bar{L})-\frac{3\epsilon}{8}\nu_s(G), \qquad (5.21)
\end{aligned}$$

where in (5.19) we used (5.15) (based on the definition of *SIG* and for an appropriate choice of $c_3(s)$), and in (5.20) we applied Lemma 12. By combining (5.17), (5.18), and (5.21), we get that $\hat{\nu_s}=(1\pm\epsilon)\nu_s(G)$ with high constant probability.

**6. Lower bounds for approximating the number of *s*-stars.** We also have matching lower bounds similar to what we had for the case of length-2 paths. For simplicity we state them for constant $s$, but they can be extended to nonconstant $s$.

THEOREM 5. *Let $s$ be a constant.*

1. *Any (multiplicative) approximation algorithm for the number of $s$-stars must perform $\Omega(\frac{n}{(\nu_s(G))^{\frac{1}{s+1}}})$ queries.*
2. *Any constant-factor approximation algorithm for the number of $s$-stars must perform $\Omega(n^{1-\frac{1}{s}})$ queries when the number of $s$-stars is $O(n^s)$.*
3. *Any constant-factor approximation algorithm for the number of $s$-stars must perform $\Omega(\frac{n^{s-\frac{1}{s}}}{(\nu_s(G))^{1-\frac{1}{s}}})$ queries when the number of $s$-stars is $\Omega(n^s)$.*

Since the constructions are very similar to those used in the proofs of items 1–3 of Theorem 3, we only describe the needed modifications in the constructions and the analysis. Here too we allow multiple edges in the constructions, and this assumption can be removed in a similar manner to the way it was dealt with in Theorem 3.

*Proof sketch of item* 1 *in Theorem* 5. For any choice of $\tilde{\nu}_s$, consider the family of all $n$-vertex graphs that each consists of a clique of size $b=\lceil\tilde{\nu}_s^{\frac{1}{s+1}}\rceil$ and an independent set of size $n-b$. The number of $s$-stars in the graph is $b\cdot\binom{b-1}{s}=\Theta(\tilde{\nu}_s)$ (recall that we assume that $s$ is a constant). However, in order to distinguish between a random graph in the family and the empty graph, it is necessary to perform a query on a vertex in the clique. The probability of hitting such a vertex in $o(\frac{n}{\nu_s^{\frac{1}{s+1}}(G)})$ queries is $o(1)$.

*Proof sketch of item* 2 *in Theorem* 5. First note that the lower bound in item 1, that is, $\mathbf{\Omega}(\frac{n}{(\nu_s(G))^{\frac{1}{s+1}}})$, is higher than the lower bound in this item, that is, $\mathbf{\Omega}(n^{1-\frac{1}{s}})$ for $\nu_s(G) < n^{1+\frac{1}{s}}$, and hence we may restrict our attention to the case that $\nu_s(G) \geq n^{1+\frac{1}{s}}$. We modify the construction of the two families of graphs from the proof of item 2 in Theorem 3 in the following manner. We start with the second family, denoted $\mathcal{G}_2$. As in the proof of item 2 in Theorem 3, each graph in this family is determined by two subsets: $S$ of a constant-size $c$ (which determines the gap between the number of $s$-stars in the two families), and $V \setminus S$. Each vertex in $S$ has $d' = \lceil (s! \cdot \tilde{\nu}_s)^{\frac{1}{s}} \rceil + s$ neighbors in $V \setminus S$, and each vertex in $V \setminus S$ has $d = \lfloor (\frac{s!\tilde{\nu}_s}{n})^{\frac{1}{s}} \rfloor$ neighbors (in $V \setminus S$ and possibly in $S$). Thus, the difference between the family $\mathcal{G}_2$ as defined here and as defined in the proof of item 2 in Theorem 3 is only in the setting of $d$ and $d'$.

The family $\mathcal{G}_1$ is also very similar to the one defined in the proof of item 2 in Theorem 3, but we perform a small modification, which slightly simplifies the analysis. Consider taking a graph in $\mathcal{G}_2$ and matching the edges between $V \setminus S$ and $S$. That is, we replace pairs of edges $(v, w)$, $(u, z)$, where $u$, $v \in V \setminus S$ and $w$, $z \in S$, by a single edge between $v$ and $u$. We shall refer to these edges as *special* edges. Note that the degree of each vertex in $V \setminus S$ remains $d$, and the set $S$ becomes an independent set. Let $\mathcal{G}_1$ be the family of graphs resulting from performing this operation on graphs in $\mathcal{G}_2$ (where the matching may be arbitrary). Observe that the number of $s$-stars in each graph $G \in \mathcal{G}_1$ satisfies

$$\nu_s(G) = (n - c) \cdot \binom{d}{s} < \tilde{\nu}_s,$$

and the number of $s$-stars in each graph $G \in \mathcal{G}_2$ satisfies

$$\nu_s(G) > c \cdot \binom{d'}{s} = c \cdot \binom{\lceil (s! \cdot \tilde{\nu}_s)^{\frac{1}{s}} \rceil + s}{s} > c \cdot \tilde{\nu}_s.$$

Given the above description, the two processes (that answer the queries of the algorithm and construct a random graph along the way) are essentially as in the proof of item 2 in Theorem 3. The only difference is in the setting of $d$ and $d'$ and in the fact that the first process also has a small probability of "hitting" a vertex in $S$ (at which point the algorithm can terminate, since the vertices in $S$ have degree 0). We also assume that the first process notifies the algorithm when a special edge is revealed (at which point the algorithm can terminate).

Consider both processes and observe that if the number of queries performed is $o(n^{1-\frac{1}{s}})$, then for both processes the probability of the event that a vertex $v$ in a query $(v, i)$ is determined to belong to $S$ is

$$o(n^{1-\frac{1}{s}}) \cdot \frac{c}{n - c - o(n^{1-\frac{1}{s}})} = o(n^{-\frac{1}{s}}).$$

The second observation is that for every $t = o(n^{1-\frac{1}{s}})$, the probability of the event that the answer of $\mathcal{P}_2$ to a query $q_t = (v, i)$ will be $(u, i')$, where $u \in S$, and similarly, that the answer of $\mathcal{P}_1$ corresponds to a special edge, is upper-bounded by

$$\frac{c \cdot d'}{(n-c) \cdot d - 2t} = \frac{c \cdot (\lceil (s! \cdot \tilde{\nu}_s)^{\frac{1}{s}} \rceil + s)}{(n-c) \cdot \left\lfloor \left( \frac{s! \tilde{\nu}_s}{n} \right)^{\frac{1}{s}} \right\rfloor - o(n^{1-\frac{1}{s}})} = O(n^{-(1-\frac{1}{s})}).$$

Hence, the probability that such an event occurs in a sequence of $o(n^{1-\frac{1}{s}})$ queries is $o(1)$.

For a neighbor query $q_{t+1} = (v, i)$, consider the probability that the answer to this query is $(u, i')$ for $u$ that has already appeared in the query-answer history. In the proof of item 2 of Theorem 3, we showed that this probability is sufficiently small. Here we do not (and cannot) make such a claim. However, given the way we modified the construction, as long as neither of the above-mentioned events occur, the distributions on the query-answer histories are identical.

*Proof sketch of item* 3 *in Theorem* 5. We modify the construction of the two families of graphs from the proof of item 3 in Theorem 3 in the following manner, where we start with the second family, $\mathcal{G}_2$. In $\mathcal{G}_2$ each graph contains a subset $S$ of $b = \lceil \frac{c \cdot 4s! \tilde{\nu}_s}{n^s} \rceil$ vertices. There is a complete bipartite graph between $S$ and $V \setminus S$, and there are $d - b$ perfect matchings between vertices in $V \setminus S$, where $d = \lfloor (\frac{s! \tilde{\nu}_s}{n})^{\frac{1}{s}} \rfloor$, so that every vertex in $V \setminus S$ has degree $d$. In order to define the first family, $\mathcal{G}_1$, we perform the same "edge-contraction" procedure as in the proof of item 2. That is, given a graph in $\mathcal{G}_2$, we replace pairs of edges between $V \setminus S$ and $S$ with single edges between vertices in $V \setminus S$. Here too we maintain the degrees of vertices in $V \setminus S$, and $S$ becomes an independent set. Observe that by the choice of $d$, the number of $s$-stars in each graph in $\mathcal{G}_1$ is upper-bounded by $\tilde{\nu}_s$. Assuming $\tilde{\nu}_s < n^{s+1}/c'$ for some sufficiently large constant $c'$, for every $G \in \mathcal{G}_2$, we have that $b < n/(2s)$ and so

$$\nu_s(G) \geq b \cdot \binom{n-b}{s} > b \cdot \binom{n(1-1/(2s))}{s} \geq \frac{c \cdot 4s! \tilde{\nu}_s}{n^s} \cdot \frac{\left( n\left(1 - \frac{1}{s}\right) \right)^s}{s!} > c \cdot \tilde{\nu}_s.$$

The processes $\mathcal{P}_1$ and $\mathcal{P}_2$ are defined very similarly to the way they were defined in the proof of item 3 in Theorem 3, where $d$ and $|S| = b$ are as defined above. Other than the different setting of the parameters, here we take into account (in the definition of $\mathcal{P}_1$) the fact that in each graph in $\mathcal{G}_1$, the $d$ perfect matchings are only between vertices in $V \setminus S$, and that there is a probability of "hitting" vertices in $S$. For both processes, if the number of queries performed is $o(n^{s-\frac{1}{s}}/{\tilde{\nu}_s}^{1-\frac{1}{s}})$, then the probability that a vertex $v$ in a query $(v, i)$ is determined to belong to $S$ is

$$o(n^{s-\frac{1}{s}}/{\tilde{\nu}_s}^{1-\frac{1}{s}}) \cdot \frac{b}{n - o(n^{s-\frac{1}{s}}/\tilde{\nu}_s^{1-\frac{1}{s}})} = o(n^{s-\frac{1}{s}}/\tilde{\nu}_s^{1-\frac{1}{s}}) \cdot \frac{c \cdot 4s! \tilde{\nu}_s / n^s}{n}$$
$$= o(\tilde{\nu}_s^{\frac{1}{s}}/n^{(1+\frac{1}{s})}) = o(1).$$

Next, for every $t = o(n^{1-\frac{1}{s}})$, the probability that the answer of $\mathcal{P}_2$ to a query $q_t = (v, i)$ will be $(u, i')$, where $u \in S$, and similarly, that the answer of $\mathcal{P}_1$ corresponds to a special edge, is upper-bounded by

$$O\left( \frac{\tilde{\nu}_s / n^s}{(\tilde{\nu}_s / n)^{\frac{1}{s}}} \right) = 0(\tilde{\nu}_s^{1-\frac{1}{s}}/n^{s-\frac{1}{s}}).$$

Therefore, the probability that such an event occurs in a sequence of $o(n^{s-\frac{1}{s}}/{\tilde{\nu}_s}^{1-\frac{1}{s}})$ queries is $o(1)$. If none of the above events occur, then we get the same distribution on query-answer histories.

**7. Other small subgraphs.** Other than $s$-stars, two additional natural extensions of length-2 paths are triangles and length-3 paths (or, more generally, length-$L$ paths).

We first observe that there are lower bounds that are linear in the number of edges $m$ when $m = \Theta(n)$, both for triangles and for length-3 paths. These lower bounds hold in the query model studied in this paper, that is, assuming the algorithm is allowed only degree queries and neighbor queries. Moreover, these lower bounds hold even if the algorithm is also allowed to sample edges uniformly. However, they do not hold if the algorithm is allowed vertex-pair queries, that is, if it may ask whether there is an edge between any two vertices $u$ and $v$ of its choice. Thus, it is possible that there are sublinear algorithms for approximating the number of these subgraphs assuming the algorithm is allowed vertex-pair queries. It can be verified that in the case of length-2 paths, and more generally, $s$-stars, the lower bounds hold even when allowed vertex-pair queries.[4]

THEOREM 6. *For $m = O(n)$, it is necessary to perform $\Omega(m)$ queries in order to distinguish with high constant probability between the case that a graph contains $\Theta(n)$ triangles and the case that it contains no triangles. This bound holds when neighbor and degree queries are allowed.*

*Proof.* Consider the following two families of graphs. In the first family each graph consists of a complete bipartite graph between two vertices and all other vertices. In the second family each graph consists of a complete bipartite graph between two vertices and all other vertices but one, where this vertex is an isolated vertex. In addition there is an edge between the two vertices. Within each family the graphs differ only in the labeling of vertices and in the labeling of the edges incident to each vertex. Observe that in both families the two high-degree vertices have degree $n-2$ and the rest of the vertices have degree 2, with the exception of the single isolated vertex in the second family. By construction, each graph in the first family contains no triangles and each graph in the second family contains $n-3$ triangles. However, in order to distinguish between a random graph in the first family and a random graph in the second family, it is necessary to either hit the isolated vertex in graphs of the second family or to hit the edge between the two high-degree vertices in graphs of the second family, or to observe all neighbors of one of the high-degree vertices in the first family. In the latter case, $n-2$ queries are necessary, and in the former cases $\Omega(n)$ queries are necessary (in order for one of these events to occur with constant probability). ∎

THEOREM 7. *For $m = O(n)$, it is necessary to perform $\Omega(m)$ queries in order to distinguish with high constant probability between the case that a graph contains $\Theta(n^2)$ length-3 paths and the case that it contains no length-3 paths. This bound holds when neighbor and degree queries are allowed.*

*Proof.* Consider the following two families of graphs, where we assume for simplicity that $n$ is even (otherwise there is an isolated vertex and the graph is defined over $n-1$ vertices, where $n-1$ is even). In the first family, each graph consists of two stars, where in each star there are $n/2$ vertices (including the center vertex). In the second family, each graph consists of two stars, where in each star there are $n/2-1$ vertices (including the center vertex). In addition, there are two isolated vertices, and there is an edge between the two star centers. Graphs in the two families differ only in the labeling of vertices and in the labeling of the edges for the star centers. Observe that in both families, the star centers have degree $n/2$. By construction, each graph in the first family contains no length-3 paths and each graph in the second family

[4]To verify this note that the lower bounds are essentially based on "hitting" a certain subset of vertices, either by querying one of these vertices or receiving one of them in an answer to a neighbor queries. If vertex-pair queries are allowed, then the algorithm still needs to hit a vertex in this subset in one of its queries.

contains $\Theta(n^2)$ length-3 paths. However, in order to distinguish between a random graph in the first family and a random graph in the second family, it is necessary to either hit one of the isolated vertices in graphs of the second family or to hit the edge between the centers in graphs of the second family, or to observe all neighbors of one of the centers in the first family. In the latter case, $n/2$ queries are necessary, and in the former cases, $\Omega(n)$ queries are necessary (in order for one of these events to occur with constant probability). □

**Acknowledgments.** We would like to thank the anonymous reviewers of SODA 2010 and the anonymous reviewers of this manuscript for their helpful comments.

## REFERENCES

[ADH+08] N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp, *Biomolecular network motif counting and discovery by color coding*, Bioinformatics, 24 (2008), pp. 241–249.

[AFS09] O. Amini, F. Fomin, and S. Saurabh, *Counting subgraphs via homomorphisms*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming, 2009, pp. 71–82.

[AG09] N. Alon and S. Gutner, *Balanced hashing, color coding and approximate counting*, in Proceedings of the 4th International Workshop on Parameterized and Exact Computation (IWPEC), 2009, pp. 1–16.

[AG10] N. Alon and S. Gutner, *Balanced families of perfect hash functions and their applications*, ACM Trans. Algorithms, 6 (2010), article 54.

[AR02] V. Arvind and V. Raman, *Approximation algorithms for some parameterized counting problems*, in Proceedings of the 13th International Symposium on Algorithms and Computation (ISAAC), 2002, pp. 453–464.

[AYZ95] N. Alon, R. Yuster, and U. Zwick, *Color coding*, J. ACM, 42 (1995), pp. 844–856.

[AYZ97] N. Alon, R. Yuster, and U. Zwick, *Finding and counting given length cycles*, Algorithmica, 17 (1997) pp. 209–223.

[BBCG08] L. Becchetti, P. Boldi, C. Castillo, and A. 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 (KDD), 2008, pp. 16–24.

[BHKK09] A. Björklund, T. Husfeldt, P. Kasaki, and M. Koivisto, *Counting paths and packings in halves*, in Proceedings of the Seventeenth Annual European Symposium on Algorithms (ESA), Lecture Notes in Comput. Sci. 5747, Springer, Berlin, 2009, pp. 578–586.

[BKKR10] I. Ben-Eliezer, T. Kaufman, M. Krivelevich, and D. Ron, *Comparing the strength of query types in property testing: The case of testing k-colorability*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2008, pp. 1213–1222.

[CEF+05] A. Czumaj, F. Ergun, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler, *Approximating the weight of the Euclidean minimum spanning tree in sublinear time*, SIAM J. Comput., 35 (2005), pp. 91–109.

[CRT05] B. Chazelle, R. Rubinfeld, and L. Trevisan, *Approximating the minimum spanning tree weight in sublinear time*, SIAM J. Comput., 34 (2005), pp. 1370–1379.

[CS09] A. Czumaj and C. Sohler, *Estimating the weight of metric minimum spanning trees in sublinear time*, SIAM J. Comput., 39 (2009), pp. 904–922.

[DLR95] R. Duke, H. Lefmann, and V. Rödl, *A fast approximation algorithm for computing the frequencies of subgraphs in a given graph*, SIAM J. Comput., 24 (1995), pp. 598–620.

[DSG+08] B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan, *QNet: A tool for querying protein interaction networks*, J. Comput. Biol., 15 (2008), pp. 913–925.

[Fei06] U. Feige, *On sums of independent random variables with unbounded variance and estimating the average degree in a graph*, SIAM J. Comput., 35 (2006), pp. 964–984.

[FG04] J. Flum and M. Grohe, *The parameterized complexity of counting problems*, SIAM J. Comput., 33 (2004), pp. 892–922.

[GK07] J. Grochow and M. Kellis, *Network motif discovery using subgraph enumeration and symmetry-breaking*, in Proceedings of the 11th Annual International Conference Research in Computational Molecular Biology (RECOMB), 2007, pp. 92–106.

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

[GR08] O. Goldreich and D. Ron, *Approximating average parameters of graphs*, Random Structures Algorithms, 32 (2008), pp. 473–493.

[GS09] M. Gonen and Y. Shavitt, *Approximating the number of network motifs*, Internet Math., 6 (2009), pp. 349–372.

[HA08] D. Hales and S. Arteconi, *Motifs in evolving cooperative networks look like protein structure networks*, Netw. Heterog. Media, 3 (2008), pp. 239–249.

[HBPS07] F. Hormozdiari, P. Berenbrink, N. Przulj, and S. C. Sahinalp, *Not all scale-free networks are born equal: The role of the seed graph in ppi network evolution*, PLoS: Computational Biology, 3 (2007), p. e118.

[KIMA04] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, *Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs*, Bioinformatics, 20 (2004), pp. 1746–1758.

[KKR04] T. Kaufman, M. Krivelevich, and D. Ron, *Tight bounds for testing bipartiteness in general graphs*, SIAM J. Comput., 33 (2004), pp. 1441–1483.

[KL83] R. Karp and M. Luby, *Monte-carlo algorithms for enumeration and reliability problems*, in Proceedings of the Twenty-Fourth Annual Symposium on Foundations of Computer Science (Annual IEEE Symposium on Foundations of Computer Science), 1983, pp. 56–64.

[Kou08] I. Koutis, *Faster algebraic algorithms for path and packing problems*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 5125, Springer, Berlin, 2008, pp. 575–586.

[KW09] I. Koutis and R. Williams, *Limits and applications of group algebras for parameterized problems*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming, 2009, pp. 653–664.

[MSOI+02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, *Network motifs: Simple building blocks of complex networks*, Science, 298 (2002), pp. 824–827.

[NO08] H. N. Nguyen and K. Onak, *Constant-time approximation algorithms via local improvements.*, in Proceedings of the Forty-Ninth Annual Symposium on Foundations of Computer Science (Annual IEEE Symposium on Foundations of Computer Science), 2008, pp. 327–336.

[PCJ04] N. Przulj, D. G. Corneil, and I. Jurisica, *Modeling interactome: Scale-free or geometric?*, Bioinformatics, 20 (2004), pp. 3508–3515.

[PR07] M. Parnas and D. Ron, *Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms*, Theoret. Comput. Sci., 381 (2007), pp. 183–196.

[SIKS06] J. Scott, T. Ideker, R. Karp, and R. Sharan, *Efficient algorithms for detecting signaling pathways in protein interaction networks*, J. Comput. Biol., 13 (2006), pp. 133–144.

[SSRS06] T. Shlomi, D. Segal, E. Ruppin, and R. Sharan, *QPath: A method for querying pathways in a protein-protein interaction network*, BMC Bioinf., 7 (2006), article 199.

[VW09] V. Vassilevska and R. Williams, *Finding, minimizing, and counting weighted subgraphs*, in Proceedings of the Fourty-First Annual ACM Symposium on the Theory of Computing, 2009, pp. 455–464.

[Wer06] S. Wernicke, *Efficient detection of network motifs*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3 (2006), pp. 347–359.

[Wil09] R. Williams, *Finding paths of length $k$ in $o^*(2^k)$ time*, Inform. Process. Lett., 109 (2009), pp. 315–318.

[YYI09] Y. Yoshida, M. Yamamoto, and H. Ito, *An improved constant-time approximation algorithm for maximum matchings*, in Proceedings of the Forty-First Annual ACM Symposium on the Theory of Computing, 2009, pp. 225–234.