From Decision to Random Certificates: Exponential Separation for Edge Estimation with Independent Set Queries

Debarshi Chanda
Indian Statistical Institute, Kolkata, India

Buddha Dev Das
Indian Statistical Institute, Kolkata, India

Arijit Ghosh
Indian Statistical Institute, Kolkata, India

Gopinath Mishra
Institute of Mathematical Sciences, Chennai, India

Abstract

We study the problem of estimating the number of edges in an undirected, unweighted graph using sublinear query access. We consider a query model that preserves the structure of Independent Set (IS) queries, but augments their output with a random certificate: given a vertex subset, the oracle returns a uniformly random edge from the induced subgraph if one exists, and returns null otherwise.

Using this access, we give a randomized algorithm that outputs a $(1 \pm \varepsilon)$-approximation to the number of edges with constant success probability using $\widetilde{O}(\log^2 m)$ queries. This implies an exponential separation from both standard IS queries and global random edge-sampling models: estimating the number of edges using standard IS queries require $\widetilde{\Theta}\left(\min\left\{\sqrt{m}, \frac{n}{\sqrt{m}}\right\}\right)$ queries, while direct random edge-sample access requires $\widetilde{\Theta}(\sqrt{m})$ samples. Beyond separation in query complexity, our algorithm is output-sensitive: its query complexity is polylogarithmic in the number of edges in the graph. This aligns with the classical objective in group testing, where one seeks algorithms that are both worst-case optimal and instance-adaptive.

Conceptually, our model connects group testing, the decision–versus–counting dichotomy, graph property testing, and the "power of a random certificate", and can be viewed as a structured form of conditional sampling of edges in graphs.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Property Testing, Edge Estimation

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.41

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/abs/2607.07483` [20]

Funding *Arijit Ghosh*: Arijit Ghosh acknowledges partial support from the Science and Engineering Research Board (SERB), Government of India, through the MATRICS grant MTR/2023/001527, and from the Department of Science and Technology (DST), Government of India, through grant TPN-104427.

Acknowledgements We thank an anonymous reviewer for their suggestions in improving the query complexity from $\log^3 m$ (in an earlier version) to $\log^2 m$.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2026).
Editors: Mohit Singh and Tom Gur; Article No. 41; pp. 41:1–41:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

We consider the problem of estimating the number of edges in an undirected, unweighted, simple graph using a sublinear number of queries. Let $G = (V, E)$ be a graph with $V = [n] := \{1, 2, \dots, n\}$ vertices and $|E| = m$ edges. Given query access to G and an approximation parameter $\varepsilon \in (0, 1)$, the objective is to output an estimate $\widehat{m}$ such that

$$(1-\varepsilon)m \;\le\; \widehat{m} \;\le\; (1+\varepsilon)m$$

with success probability at least $2/3$.

Query model and our result

We work in a query model built on *Independent Set* (IS) queries, introduced by [6]. The IS query takes a subset of vertices $U \subseteq V$ and returns whether U is an independent set.

We consider the *Independent Set with Certificate* query, denoted by ISC, which takes as input a subset of vertices $U \subseteq V$ and returns

$$\texttt{ISC}(U) = \begin{cases} e \sim \mathrm{Unif}(E(U)), & \text{if } E(U) \neq \emptyset, \\ 0, & \text{otherwise,} \end{cases}$$

where $E(U)$ denotes the set of edges in the subgraph of G induced by U, and $\mathrm{Unif}(S)$ denotes the uniform distribution over a (nonempty) set S.

Importantly, the ISC query preserves the structural constraint of a standard IS query: the algorithm specifies exactly the same vertex subsets as in the IS model, and the only difference lies in the response, which returns a uniformly random edge as a certificate whenever the induced subgraph is nonempty.

[6] showed that the number of edges in a graph can be estimated using

$$\widetilde{O}\left(\min\left\{\frac{n^2}{m}, \sqrt{m}\right\}\right) \;=\; \widetilde{O}(n^{2/3})$$

IS queries, establishing a polynomial separation from local query based algorithms that require $\Theta(n)$ queries in the worst case [27]. In a subsequent work, [21] nearly resolved the query complexity of edge estimation using IS queries by showing that

$$\widetilde{\Theta}\left(\min\left\{\frac{n}{\sqrt{m}}, \sqrt{m}\right\}\right)$$

queries are both necessary and sufficient. In particular, this implies a worst-case complexity of $\widetilde{\Theta}(\sqrt{n})$ IS queries.

Our main result shows that the natural enrichment of IS queries with random certificates yields an exponential separation from standard IS queries for edge estimation, achieving polylogarithmic query complexity in m.

▶ **Theorem 1** (Main result). *There exists a randomized algorithm that, given access to a graph G through ISC queries and an approximation parameter $\varepsilon \in (0, 1/3)$, outputs an estimate $\widehat{m}$ of the number of edges m in G such that, with probability at least $2/3$,*

$$(1-\varepsilon)m \;\le\; \widehat{m} \;\le\; (1+\varepsilon)m.$$

The algorithm uses $O(\varepsilon^{-2}\log^2 m \cdot \log\log m)$ ISC queries in expectation.

The query complexity of the algorithm is *strictly output-sensitive* in the sense that it depends only on the number of edges, with no dependence on n, even in lower-order terms.

Theorem 1 not only demonstrates the power gained by augmenting IS access with random certificates over standard IS access, but also underscores the strength of sampling within induced subgraphs. When edges can only be sampled uniformly at random from the entire graph, estimating the number of edges requires $\widetilde{\Theta}(\sqrt{m})$ samples [30]. In contrast, allowing the algorithm to sample a uniformly random edge from a queried induced subgraph yields an exponential separation from global random edge-sampling, achieving query complexity $\widetilde{O}(\log^2 m)$.

In what follows, we highlight conceptual connections between our results and query model with several well-studied themes, including group testing, graph property testing, the power of random certificates, decision versus counting problems, and conditional sampling.

Group testing perspective

The introduction of IS queries in graph algorithms was originally motivated by ideas from *group testing*. In classical group testing, introduced by [25] and studied extensively since then, the goal is to infer the size of an unknown set of defective items by querying subsets of a known universe [3]. A standard group test returns only a binary result indicating whether the queried subset contains at least one defective item.

This viewpoint applies naturally to graph edge estimation. Here, edges play the role of defective items, and the allowed vertex pairs for queries impose structural constraints on the possible subsets of edges that can be queried. [6] introduced both IS and Bipartite Independent Set (BIS) queries through this lens. A BIS query takes two disjoint vertex sets as input and returns whether there exists an edge with one endpoint in each set.

In the same vein, [13] and [5] introduced the Edge Emptiness query EE to study triangle estimation and connectivity in graphs, respectively. An EE query specifies a set of candidate edges and reports whether at least one of them exists in the graph.

Decision versus counting

In a series of seminal works, [32, 33] studied the relationship between approximate counting and decision access for Boolean formulas. In particular, they showed that given decision access to satisfiability, one can approximately count the number of satisfying assignments of any Boolean formula on n variables using only $O(\log n)$ such queries.

Our work establishes an analogous phenomenon in the setting of graph edge estimation: augmenting decision-style IS queries with a random certificate enables efficient approximate counting of edges in a graph.

Conditional sampling

Our model also bears a close conceptual relationship to work on property testing and estimation with conditional sampling access. In distribution testing, a conditional sampling oracle [18, 16] allows the algorithm to specify a subset of the domain and receive a sample drawn from the distribution restricted to that subset. Such structured sampling can lead to exponential savings over standard sampling in tasks such as uniformity testing, support size estimation, and testing of joint distributions [15, 9, 14, 10].

In our setting, the ISC oracle plays an analogous role: the algorithm specifies a vertex subset, and if the induced subgraph is nonempty, it receives a uniformly random edge from that subgraph. Thus, ISC can be viewed as a structured conditional sampling primitive over the edge set of the graph, where the conditioning is constrained to induced subgraphs. Our results demonstrate that, as in distribution testing, such conditional access can fundamentally change the complexity of estimation problems in graphs.

Power of a random certificate

Studying the power of random certificates for approximate counting is an important question across a variety of query models [31, 17]. These works show that augmenting group queries with access to random certificates can lead to significant reductions in query complexity.

In the graph setting, several queries have been introduced from the viewpoint of group queries while retaining the structural properties of the graph. Two notable examples are `IS` and `BIS`, introduced by [6] for estimating the number of edges. Our `ISC` query fits naturally within this framework: it preserves the structure of `IS` while adding a random certificate. In particular, our algorithm achieves substantially lower query complexity compared to algorithms using only `IS` access, and also improves over the best known $\widetilde{O}(\log^5 n)$-query algorithm based on `BIS` access due to [2].

The main difference between the power of random certificates studied in [31, 17] and our setting is that our query oracles are group graph queries that retain the structural properties of the underlying graph.

Graph property testing

Estimating the number of edges in a graph with sublinear queries has been studied in several access models. Early works focused on local queries such as degree, neighbour, and pair queries [28, 29, 27, 34], which provide fine-grained access but face inherent limits for edge estimation.

This led to models that incorporate global information. In particular, augmenting local access with random edge sampling [4, 26, 7] enables more efficient estimation. More recently, a finer-grained analysis of edge estimation under local queries enriched with random edge sampling have provided an understanding of the tradeoffs between local and global access [11, 12, 19, 7].

Another line of work studies structured global queries, notably `IS` and `BIS` [6, 21, 2]. These queries provide decision-only access over vertex subsets while preserving graph structure, and have been extended to hypergraphs [23, 8, 24]. We discussed their known upper and lower bounds earlier in the Introduction.

Finally, [1] showed that augmenting `IS` queries with additional local queries yields a quadratic separation from using `IS` queries alone, further illustrating the power of combining different query primitives. Other than the `BIS` query, all query models discussed above requires polynomial number of queries in the worst case.

Notations

Throughout, the notation $\widetilde{O}(f)$, suppress factors polynomial in $1/\varepsilon$ and $\log f$. For an integer k, we define $[k] = \{1, \dots, k\}$. We write $a = (1 \pm \varepsilon)b$ to indicate that a is a $(1 \pm \varepsilon)$-approximation of b.

Throughout, we use $G = (V, E)$ to denote the input graph with $m = |E|$ number of edges and $V = |V|$ vertices. For a subset $U \subseteq V$, let $m(U)$ be the number of edges in the graph induced by the subset U.

2 Technical overview

We give a high-level description of our algorithm for estimating the number of edges in an undirected graph using Independent Set with Certificate (ISC) queries. Conceptually, our approach converts approximate counting into a sequence of controlled decisions and sampling steps that gradually reduce the graph while preserving a constant fraction of its edges.

A central conceptual contribution is the use of *approximation preserving subsets* and *approximation preserving sequences*, which formalise how much of the graph's edge mass is retained at each stage and guide both the algorithm and its analysis. We leverage graph sparsification techniques to enable iterative construction of approximation-preserving sequences on graphs using ISC queries.

Outline of the Algorithm

At a high level, our algorithm maintains a nested sequence of vertex subsets

$$V = U_0 \supseteq U_1 \supseteq \cdots \supseteq U_\kappa,$$

where each U_j contains a constant fraction of the edges of U_{j-1}. The algorithm alternates between two modes:

- If the current induced subgraph is "small" (that is, has fewer than τ edges), we estimate its edge count directly.
- Otherwise, we construct a smaller approximation preserving subset of the graph by randomly partitioning the vertex set and selecting a part that provably retains a constant fraction of the edges.

This iterative reduction of the graph continues until we reach a subset U_κ whose edge count is small enough to estimate accurately; we set the threshold for U_κ to be a sufficiently large constant. The final estimate of m is then reconstructed by multiplying the relative edge losses across all steps.

Where the $\widetilde{O}(\log^2 m)$ query complexity comes from

A key novelty of our approach is that all complexity bounds are driven by $\log m$ rather than $\log n$. This is achieved through three ideas that are not present in prior IS or BIS based algorithms:

- *Random certificates enable output-sensitive sparsification:* Unlike decision-only IS or BIS queries, ISC queries allow us to estimate how many edges of a subset lie in a randomly chosen part, which lets us iteratively construct smaller induced subgraphs to work on while tracking edge mass locally.
- *Approximation Preserving Sequence:* Instead of estimating m directly, we construct a chain of subsets $U_0 \supseteq U_1 \supseteq \cdots \supseteq U_\kappa$ such that each U_j retains a constant fraction of edges of U_{j-1}. Crucially, we establish κ to be $O(\log m)$ in expectation, independent of n. We also bound the number of queries required to construct this sequence as $O(\kappa^2)$ and bound its expectation.
- *Local multiplicative reconstruction:* We only need to estimate the ratios $p_j = \frac{m(U_j)}{m(U_{j-1})}$, each of which is $\Omega(1)$ due to the $U_0, \ldots, U_\kappa$ being an approximation preserving subsequence. This keeps the per-level query complexity small and makes the overall cost in expectation $\widetilde{O}(\kappa^2) = \widetilde{O}(\log^2 m)$.

Estimating Small Edge Counts

Fix a threshold τ. One can design an algorithm that makes $O(\sqrt{\tau})$ `ISC` queries and guarantees the following with desired probability about a given subset $U \subseteq V$. If $m(U) \leq \tau$, then the algorithm reports an approximation of $m(U)$. Moreover, if $m(U) \geq 8\tau$, the algorithm reports that $m(U)$ is large. The proof is based on the birthday paradox. The details can be found in Section 3.

Graph Sparsification

The idea here is to leverage graph sparsification to construct a smaller subset of vertices to work on at each step. We partition U uniformly at random into two subsets U^1 and U^2. A graph sparsification lemma, along the result of [6], shows that with desired probability,

$$\left|\frac{m(U)}{2} - \sum_{i=1}^{2} m(U^i)\right| \leq c\sqrt{m(U)}\log m(U).$$

We provide the proof in Section 4. Intuitively, this means that the total number of edges inside U^1 and U^2 is close to half of $m(U)$, up to a lower-order error term. As long as $m(U)$ is sufficiently large, this implies that at least one of U^1 or U^2 must contain a constant fraction of the edges of U. More precisely, there exists an absolute constant ζ_{sp} such that if $m(U) \geq \zeta_{\mathrm{sp}}$, then with probability at least $1 - 1/m(U)$, at least one of $m(U^1)/m(U)$ and $m(U^2)/m(U)$ is at least $1/8$, and both $m(U^1)/m(U)$ and $m(U^2)/m(U)$ are at most $3/4$.

Choosing the Good Part

After partitioning U into U^1 and U^2, we must identify which part retains an appropriate fraction of the edges. Note that this step is fundamentally enabled by the `ISC` queries. The idea is to take a random sample of the edges in U, and check whether it is preserved in the induced subgraph of either U^1 or U^2.

To do this, we draw random edges from U via `ISC` queries. Let $\widehat{p}(U^i)$ be the fraction of sampled edges that fall inside U^i, for $i \in [2]$. If $\widehat{p}(U^i) \geq 1/16$, we select U^i as the next subset; otherwise, we pick an arbitrary part. A Chernoff bound guarantees that, with desired probability, we correctly identify a subset satisfying

$$\frac{1}{32} \leq \frac{m(U^i)}{m(U)} \leq \frac{3}{4}.$$

Such subsets are called *approximation preserving subsets*. The upper bound of $3/4$ is useful to bound the final query complexity and the lower bound of $1/32$ is useful for guaranteeing the quality of the estimator.

Constructing an Approximation Preserving Sequence

Starting with $U_0 = V$, we iterate:
1. Run threshold checking on U_j.
2. If $m(U_j) < \tau$, stop and set $U_\kappa = U_j$.
3. Otherwise, sparsify U_j and select an approximation preserving subset U_{j+1}.

This yields a sequence $V = U_0 \supseteq U_1 \supseteq \cdots \supseteq U_\kappa$, where, with probability at least $4/5$, each U_j satisfies

$$\frac{1}{32} \leq \frac{m(U_j)}{m(U_{j-1})} \leq \frac{3}{4}.$$

Final Estimation

For each $j \in [\kappa]$, define

$$p_j = \frac{m(U_j)}{m(U_{j-1})}.$$

Then we have the identity

$$m = \frac{m(U_\kappa)}{\prod_{j=1}^{\kappa} p_j}.$$

Instead of estimating each $m(U_j)$ from scratch, we estimate only the ratios p_j. Because each $p_j = \Omega(1)$ by construction of the approximation preserving sequence, we can estimate each p_j using small number of queries, independent of n or $m(U_j)$. This yields an estimate $\widehat{p}_j$ satisfying the following with desired probability.

$$\widehat{p}_j = (1 \pm \varepsilon/O(\sqrt{\kappa}))\, p_j$$

Note that the entire approximation preserving subsequence is constructed a priori. Hence, we know κ during the final estimation phase that we use to fix the approximation parameters appropriately for the multiplicative estimation for each $p_j, j \in [\kappa]$. Let $\widehat{m}(U_\kappa)$ be the estimate obtained via the birthday-based estimator. Our final output is

$$\widehat{m} = \frac{\widehat{m}(U_\kappa)}{\prod_{j=1}^{\kappa} \widehat{p}_j}.$$

A careful multiplicative error analysis shows that this is a $(1 \pm \varepsilon)$-approximation to m.

Proof Roadmap

The correctness proof proceeds in the following stages:

- *Threshold correctness:* We show that the collision-based test reliably distinguishes between $m(U) < \tau$ and $m(U) \geq 8\tau$, and that the birthday-paradox estimator is accurate when $m(U) < \tau$.
- *Constructing approximation preserving subsequence:* While we decrease the failure probability of choosing a good subset at each iteration, the probability of failure of the sparsification step depends on the number of edges remaining. This necessitates a careful analysis of the failure probability of the sparsification step and of our threshold-checking mechanisms to establish the correctness of our algorithm and query complexity bounds.
- *Multiplicative reconstruction:* Given our approximation preserving subsequence, we estimate each p_j with appropriately scaled approximation error. Then, we bound the cumulative error from estimating all p_j and show that the product estimator preserves a global $(1 \pm \varepsilon)$ approximation.
- *Complexity analysis:* We show that $\mathbb{E}[\kappa] = O(\log m)$ and that the total query complexity is dominated by estimating the p_j's, yielding an overall expected complexity of $\widetilde{O}(\log^2 m/\varepsilon^2)$.

Outline

We begin in Section 3, where we present the threshold-checking algorithm and show how to estimate the edge count when the number of edges is below a given threshold. In Section 4, we describe the algorithm for constructing an approximation-preserving subsequence. In Section 4.1, we establish how graph sparsification yields an approximation-preserving subset, and use this result in Section 4.2 to construct an approximation-preserving subsequence. Finally, in Section 5, we present the complete algorithm for edge estimation. Appendix A discusses some useful concentration inequalities.

3 Threshold checking

Let a fixed threshold ($\tau \geq 1$) be given. We consider the following threshold-checking problem: given a subset of vertices $U \subseteq V$, determine whether $m(U) < \tau$. If this condition holds, the goal is to estimate the $(1 \pm \varepsilon)$ approximation of $m(U)$.

To address this task, we present two algorithms `IsCollision` and `BirthdayEstimate` based on the birthday paradox techniques [30, Chapter 5]. `IsCollision` distinguishes between the cases $m(U) < \tau$ or $m(U) \geq 8\tau$ (Lemma 2). `BirthdayEstimate` outputs an $(1 \pm \varepsilon)$ approximation of $m(U)$, if $m(U) < \tau$ (Lemma 3). The corresponding guarantees are stated below. For details, see the full version of the paper [20].

▶ **Lemma 2** (Correctness of `IsCollision`). *For a subset $U \subseteq V$ and a threshold $\tau \geq 1$, the following statements hold for `IsCollision` with probability at least $1 - \delta$:*

(i) *If $m(U) \leq \tau$, then `IsCollision` outputs 1.*

(ii) *If $m(U) \geq 8\tau$, then `IsCollision` outputs 0.*

The number of `ISC` queries made is $O(\sqrt{\tau}\log(\delta^{-1}))$

Suppose that a collision is detected, i.e., `IsCollision` outputs 1. We can obtain a $(1 \pm \varepsilon)$-approximation of $m(U)$ using additional `ISC` queries. This estimation procedure is implemented in `BirthdayEstimate`.

▶ **Lemma 3** (Correctness of `BirthdayEstimate`). *For a subset $U \subseteq V$, threshold $\tau \geq 1$ and approximation error $\varepsilon > 0$. If $m(U) \leq \tau$, then `BirthdayEstimate` outputs an $(1 \pm \varepsilon)$ estimate of $m(U)$ with probability at least $15/16$. The number of `ISC` query made is $\Theta\left(\varepsilon^{-1}\sqrt{\tau}\right)$.*

4 Sparsification

In this section, we will describe a procedure to generate an approximation preserving sequence (formally defined below) $U_0, \ldots, U_\kappa$ with probability at least $7/8$. Moreover, the expected query complexity of the procedure is $O(\log^2 m)$.

▶ **Definition 4** (Approximation preserving subset). *For $U, U' \subset V$ such that $U' \subseteq U$, we say that U' is an* approximation preserving subset *of U if*

$$\frac{1}{32} \leq \frac{m(U')}{m(U)} \leq \frac{3}{4}.$$

▶ **Definition 5** (Approximation preserving sequence). *For a sequence $U_0, U_1, \ldots, U_\kappa \subseteq V$ with $U_0 \supseteq U_1 \supseteq \ldots \supseteq U_\kappa$, We say that $U_0, U_1, \ldots, U_\kappa$ is an* approximation preserving sequence *if U_{i-1} is an approximation preserving subset of U_i for all $i \in [\kappa]$.*

Formally, we will prove the following lemma in this section.

▶ **Lemma 6** (Approximate sequence lemma). *There exists an algorithm that outputs an approximation preserving sequence $U_0, U_1, \ldots, U_\kappa$ with probability at least $4/5$. The expected value of κ is $O(\log m)$, and the expected number of `ISC` queries made is $O(\log^2 m)$.*

To prove the above lemma, we start with $U_0 = V$, and find U_i iteratively. Informally speaking, we stop with U_κ when $m(U_\kappa)$ is *small enough*. In Section 4.1, we discuss a procedure (`ApproxSubset` corresponding to Lemma 7) that generates a U' which is an approximation preserving subset of U, with specified probability, for any given U as long as $m(U)$ is *large*. In Section 4.2, we describe the procedure (`ApproxSequence` corresponding to Lemma 12) that uses `IsCollision` and `ApproxSubset` iteratively to generate an approximation preserving subsequence $U_0, \ldots, U_\kappa$. Essentially, `IsCollision` is used to determine whether the number of edges in the current subset U_i is small or large and `ApproxSubset` is used to generate a good subset as long as $m(U_i)$ is large.

4.1 Procedure to find approximate preserving subset

In this section, we describe the procedure `ApproxSubset` (see Algorithm 1) for finding the approximation preserving subset $U' \subseteq U$, for any given U and prove its guarantee.

Algorithm 1 `ApproxSubset`(U, δ).

Require: `ISC` query access to G and a confidence parameter $\delta \in (0, 1)$.
Ensure: A subset $U' \subseteq U$.
1: Uniformly and independently partition U into U^1 and U^2.
2: Uniformly and independently sample $O(\log(\delta^{-1}))$ random edges from U and let S be the multi-set of sampled edges.
3: Compute $\widehat{p}(U^i) = \frac{1}{|S|}\sum_{e\in S} \mathbf{1}_{e\in U^i}$ for $i \in \{1, 2\}$.
4: **if** There exists $i \in \{1, 2\}$ such that $\widehat{p}(U^i) \geq \frac{1}{16}$ **then**
5:     **return** U^i as U'
6: **else**
7:     **return** U^1 or U^2 arbitrarily as U'.
8: **end if**

Formally, we will prove the following.

▶ **Lemma 7** (Approximate subset lemma). *Let ζ_{sp} be a suitably large constant. Consider* `ApproxSubset` *that takes $U \subseteq V$ as input and produces $U' \subseteq U$ as the output.*
(i) $\mathbb{E}[m(U')] \leq m(U)/2$;
(ii) *If $m(U) \geq \zeta_{\mathrm{sp}}$, with probability $1 - \delta - 1/m(U)$, U' is an approximation preserving subset of U.*

The construction of an approximation-preserving subset U' proceeds in two steps. First, we sparsify the graph by randomly partitioning the set U into U^1 and U^2 (see Lemma 8). When $m(U)$ is large enough, this partition guarantees the existence of U^i such that $m(U^i)$ is a constant fraction of $m(U)$ (Corollary 10). In the second step, we identify such an approximation-preserving subset from U^1 and U^2 (Lemma 11). Combining these results yields the proof of Lemma 7.

We prove the following variant of the graph sparsification result of Beame et al. [6].

▶ **Lemma 8** (Graph Sparsification). *For a given $\delta \in (0, 1)$, there exists an absolute constant c such that the following holds. Let $G = (V, E)$ be a graph with n vertices and m edges. Let U^1 and U^2 be a uniformly random partition of V. Then,*
(i) $\mathbb{E}[m(U^1) + m(U^2)] = m(U)/2$;
(ii) $\Pr\left[\left|m/2 - \sum_{i=1}^{2} m(U^i)\right| \geq c\sqrt{m}\log(m/\delta)\right] \leq \delta$.

We will need the following lemma to prove the above graph sparsification lemma.

▶ **Lemma 9.** *Let A be a set of ℓ elements. For every element $t \in A$, one assigns a color independently and uniformly from $\{1,2\}$. Let $C^{(i)}$ be the number of vertices in A with color i, for any $i \in \{1,2\}$. Then for $\delta \in (0,1)$, we have*

$$\Pr\left[\left|C^{(1)} - C^{(2)}\right| \geq \sqrt{6\ell \log(4\delta^{-1})}\right] \leq \delta \text{ and } \mathbb{E}\left[\left|C^{(1)} - C^{(2)}\right|\right] \leq \sqrt{\ell}$$

Proof. For $t \in A$, let X_t be the indicator random variable that is 1 if t is assigned color 1 and 0 otherwise. Then, $C^{(1)} = \sum_{t \in A} X_t$ and $\mathbb{E}\left[C^{(1)}\right] = \ell/2$. Using Chernoff's inequality (Lemma 19) with $\varepsilon = \sqrt{(6/\ell)\log(4/\delta)}$, we have

$$\Pr\left[\left|C^{(1)} - \frac{\ell}{2}\right| > \frac{\varepsilon \ell}{2}\right] \leq 2\exp\left(\frac{6\log(4\delta^{-1})}{\ell} \cdot \frac{\ell}{6}\right) \leq \frac{\delta}{2} \tag{1}$$

Similarly, we can have the same bound for color 2.

$$\Pr\left[\left|C^{(2)} - \frac{\ell}{2}\right| > \frac{\varepsilon \ell}{2}\right] \leq \frac{\delta}{2} \tag{2}$$

Observe that $\left|C^{(1)} - C^{(2)}\right| \leq \left|C^{(1)} - (\varepsilon\ell/2)\right| + \left|C^{(2)} - (\varepsilon\ell/2)\right|$. Therefore

$$\Pr\left[\left|C^{(1)} - C^{(2)}\right| > \varepsilon \ell\right] \leq \delta$$

For $t \in A$, let Y_t be a random variable that is 1 if t is assigned the color 1 and -1 otherwise. Then $C^{(1)} - C^{(2)} = \sum_{t \in A} Y_t = Y$. We have that $\Pr[Y_t = 1] = \Pr[Y_t = 1] = 1/2$, then

$$\mathbb{E}[Y_t] = 0 \text{ and } \mathbb{E}[Y_t^2] = 1$$

By the independence of Y_t, we get

$$\mathbb{E}[Y^2] = \sum_{t \in A} \mathbb{E}[Y_t^2] + \sum_{u \neq v} \mathbb{E}[Y_u Y_v] = \ell$$

Finally, we have $\mathbb{E}[|Y|] \leq \sqrt{\mathbb{E}[|Y|^2]} = \sqrt{\mathbb{E}[Y^2]} \leq \sqrt{\ell}$. ◀

We now prove Lemma 8.

Proof of Lemma 8. Let each vertex $t \in V$ be associated with a random variable $Y_t \in \{1,2\}$ such that $\Pr[Y_t = 1] = 1/2$. For $i \in \{1,2\}$, define $U^i = \{t \in V \mid Y_t = i\}$ and

$$f(Y_1, \ldots, Y_n) = m(U^1) + m(U^2).$$

A fixed edge is counted by f if and only if both its endpoints are assigned the same label, which occurs with probability 1/2. Therefore, $\mathbb{E}[f] = m/2$.

Consider the Doob martingale $X_0, X_1, \ldots, X_n$, where $X_t = \mathbb{E}[f(Y_1, \ldots, Y_n)|Y_1, \ldots, Y_t]$. For a fix values of $Y_1, \ldots, Y_{t-1}$, we have,

$$X_{t-1} = \frac{g(1) + g(2)}{2}, \text{ where } g(i) = \mathbb{E}[f(Y_1, \ldots, Y_n)|(Y_1, \ldots, Y_{t-1}) \cap (Y_t = i)]$$

Since, X_t is either $g(1)$ or $g(2)$, we have $|X_t - X_{t-1}| \leq |g(1) - g(2)|$. Let $N(t)$ be the set of neighbours of the vertex t and $\deg(t) = |N(t)|$. Let $N_{<t} = N(t) \cap [t-1]$ and

$N_{>t} = N(t) \cap \{(t+1), \ldots, n\}$. Let $C^{(i)}_{<t}$ and $C^{(i)}_{>t}$ be the number of vertices in $N_{<t}$ and $N_{>t}$ respectively. So,

$$\Delta_t = |g(1) - g(2)| = \left| C^{(1)}_{<t} + \mathbb{E}\left[C^{(1)}_{>t} \mid Y_t = 1\right] - C^{(2)}_{<t} - \mathbb{E}\left[C^{(2)}_{>t} \mid Y_t = 2\right] \right|$$

The above equation is true because any edge with vertices in $[t-1]$ will have the same contribution to $g(1)$ and $g(2)$. Similarly, any edge which have one vertex in $[t-1]$ and another in $\{(t+1), \ldots, n\}$ or both of its vertices in $\{(t+1), \ldots, n\}$ will have the same contribution. Using Lemma 9, we have with probability at least $1 - \delta'$,

$$\Delta_t \leq \sqrt{6\deg(t) \log \frac{4}{\delta'}} + \sqrt{\deg(t)} \leq 2c\sqrt{\deg(t) \log \frac{1}{\delta'}} = c_t$$

Let $\mathcal{B}$ be the event that there exits a t such that $\Delta_t > c_t$. Let n_a be the set of active vertices (vertices with degree at least 1). Therefore, using union bound $\Pr[\mathcal{B}] \leq 4n_a\delta' \leq 4m\delta'$.

Let $S = \sum_{t=1}^{n} c_t^2 = \sum_{t=1}^{n} 4c^2 \deg(t) \log(4/\delta') = O(m \log(1/\delta'))$ and $s = c\sqrt{m} \log(1/\delta')$. Using Lemma 21, we have

$$\Pr\left[\left|f - \frac{m}{2}\right| > s\right] \leq 2\exp\left(-\frac{s^2}{2S}\right) + \Pr\left[\mathcal{B}\right] \leq \delta' + 4m\delta'$$

Replacing δ' with δ/cm for a sufficiently large constant, we obtain

$$\Pr\left[\left|\frac{m}{2} - \sum_{i=1}^{2} m(U^i)\right| \geq c\sqrt{m} \log\left(\frac{m}{\delta}\right)\right] \leq \delta$$

which completes the proof. ◀

▶ **Corollary 10.** *For a subset $U \subseteq V$, suppose there exists a sufficiently large absolute constant ζ_{sp} such that $m(U) \geq \zeta_{\mathrm{sp}}$. Let U^1 and U^2 be a uniformly random partition of U. Then, with probability at least $1 - \frac{1}{m(U)}$, the following hold.*

(i) $\frac{m(U^1)}{m(U)} \geq 1/8$ *or* $\frac{m(U^2)}{m(U)} \geq 1/8$;

(ii) $\frac{m(U^1)}{m(U)} \leq 3/4$ *and* $\frac{m(U^2)}{m(U)} \leq 3/4$.

Proof. Taking ζ_{sp} to be sufficiently large, we have $c\sqrt{m(U)} \log m(U) \leq \frac{m(U)}{4}$. So, by taking $\delta = 1/m(U)$ in Lemma 8, we

$$\Pr\left[\frac{m(U)}{4} \leq \sum_{i=1}^{2} m(U^i) \leq \frac{3\,m(U)}{4}\right] \geq 1 - \frac{1}{m(U)}\,.$$

This implies that, with probability at least $1 - 1/m(U)$, at least one of $m(U^1)/m(U)$ and $m(U^2)/m(U)$ is at least $1/8$, and both $m(U^1)/m(U)$ and $m(U^2)/m(U)$ are at most $3/4$. ◀

To prove the guarantees of `ApproxSubset` described in Algorithm 1, we need Lemma 11.

▶ **Lemma 11.** *With probability at least $1-\delta$, the following statements hold for* `ApproxSubset`

(i) *If* $\frac{m(U^i)}{m(U)} \geq \frac{1}{8}$, *then* $\widehat{p}(U^i) \geq \frac{1}{16}$.

(ii) *If* $\frac{m(U^i)}{m(U)} < \frac{1}{32}$, *then* $\widehat{p}(U^i) < \frac{1}{16}$.

The number of `ISC` *queries made is* $O\left(\log(\delta^{-1})\right)$.

Proof. Consider a subset $U^i \subseteq U$ for any $i \in \{1,2\}$. For each sampled edge, define the indicator random variable

$$X_j = \begin{cases} 1 & \text{if the } j^{\text{th}} \text{ sampled edge lies in } U^i, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $\mathbb{E}[X_j] = \frac{m(U^i)}{m(U)}$.

Define the fraction of sampled edges that belong to U^i by $X = \frac{1}{|S|}\sum_{j\in[|S|]} X_j$. where S is the multiset of sampled edges (Line 2) obtained via `ISC` queries on U. Note that $X = \widehat{p}(U^i)$ and hence

$$\mathbb{E}[\widehat{p}(U^i)] = \frac{1}{|S|}\sum_{j\in[|S|]} \mathbb{E}[X_j] = \frac{m(U^i)}{m(U)} = \frac{m(U^i)}{m(U)}.$$

Applying Hoeffding's bound (Lemma 20), we get

$$\Pr\left(\left|\widehat{p}(U^i) - \frac{m(U^i)}{m(U)}\right| \geq \tfrac{1}{32}\right) \leq \exp\left(-\frac{|S|}{2}\cdot(1/32)^2\right) \leq \delta/2.$$

The last inequation holds since $|S| = O(\log 1/\delta)$. Using the union bound over $i \in \{1,2\}$, we get the desired result. ◀

We now give the proof of Lemma 7, using Lemma 8, Corollary 10 and Lemma 11

Proof of Lemma 7. In `ApproxSubset`, the set U is partitioned uniformly at random into U^1 and U^2. From Lemma 8(i),

$$\mathbb{E}[m(U^1) + m(U^2)] = m(U)/2.$$

Since U' is either U^1 or U^2, it follows that $\mathbb{E}[m(U')] \leq m(U)/2$, proving part (i).

We now prove part (ii). Let $\mathcal{E}_{\text{apx}}$ denote the event that U' is an approximation-preserving subset of U, that is, $1/32 \leq m(U')/m(U) \leq 3/4$. Let $\mathcal{E}$ be the event that at least one of $m(U^1)/m(U)$ and $m(U^2)/m(U)$ is at least $1/8$, and both $m(U^1)/m(U)$ and $m(U^2)/m(U)$ are at most $3/4$. By Corollary 10,

$$\Pr[\mathcal{E}] \geq 1 - \frac{1}{m(U)}.$$

Let us condition on the event $\mathcal{E}$. It suffices to bound $\Pr[\mathcal{E}_{\text{apx}} \mid \mathcal{E}]$. Then both $m(U^1)/m(U)$ and $m(U^2)/m(U)$ are at most $3/4$, and at least one of $m(U^1)/m(U)$ and $m(U^2)/m(U)$ is at least $1/8$; w.l.o.g. assume $m(U^1)/m(U) \geq 1/8$. If $m(U^2)/m(U) \geq 1/32$, then since $U' \in \{U^1, U^2\}$ we immediately have $1/32 \leq m(U')/m(U) \leq 3/4$, that is, $\Pr[\mathcal{E}_{\text{apx}} \mid \mathcal{E}] = 1$. Otherwise $m(U^2)/m(U) < 1/32$. By Lemma 11, with probability at least $1-\delta$, we obtain $\widehat{p}(U^1) \geq 1/16$ and $\widehat{p}(U^2) < 1/16$, implying $U' = U^1$. Consequently $1/32 \leq m(U')/m(U) \leq 3/4$ holds with probability at least $1-\delta$ conditioned on $\mathcal{E}$. Thus in all cases

$$\Pr[\mathcal{E}_{\text{apx}} \mid \mathcal{E}] \geq 1 - \delta.$$

Therefore,

$$\Pr[\mathcal{E}_{\text{apx}}] \geq \Pr[\mathcal{E}] \cdot \Pr[\mathcal{E}_{\text{apx}} \mid \mathcal{E}] \geq 1 - \frac{1}{m(U)} - \delta.$$ ◀

Algorithm 2 `ApproxSequence`(V).

Require: `ISC` access to graph G.
Ensure: A nested sequence $U_1, U_2, \cdots, U_\kappa$.

```
Initialize U_0 ← V; coll ← 1; κ ← 0
while true do
    coll ← IsCollision(U_κ, ζ_sp, 1/2^{κ+5})
    if coll equals 1 then
        break
    else
        U_{κ+1} ← ApproxSubset(U_κ, 1/2^{κ+5})
        κ ← κ + 1
    end if
end while
return U_1, U_2, ··· , U_κ
```

4.2 Procedure to find approximate preserving sequence

We now describe the algorithm `ApproxSequence` (see Algorithm 2), which constructs the sparsification sequence used by the main estimator.

▶ **Lemma 12** (Correctness of `ApproxSequence`). *Consider `ApproxSequence` as described in Algorithm 2. With probability at least 4/5, it outputs an approximation preserving sequence $U_1, U_2, \ldots, U_\kappa$ such that $m(U_\kappa) \le 8\zeta_{\text{sp}}$. Here ζ_{sp} is given large constant. The number of `ISC` queries made is $O(\kappa^2)$.*

Proof.

Correctness. Let us consider the event $\mathcal{E}_{\text{col}}$ that denotes

- There does not exist a j such that the algorithm goes to the next iteration if $m(U_j) \le \zeta_{\text{sp}}$.
- For any j such that the algorithm goes to the next iteration if $m(U_j) \ge 8\zeta_{\text{sp}}$.

The probability of $\mathcal{E}_{\text{col}}$ is lower bounded by `IsCollision` succeeds in all iterations during the run of the algorithm. Using the union bound, we have

$$\Pr[\mathcal{E}_{\text{col}}] \ge 1 - \sum_{i=1}^{\infty} \frac{1}{2^{i+5}} \ge 1 - \frac{1}{2^5} \sum_{i=1}^{\infty} \frac{1}{2^i} \ge 1 - \frac{1}{16} = \frac{15}{16}.$$

Let us continue the argument under the conditional space that $\mathcal{E}_{\text{col}}$ has occurred.

We assume $m(U_0) = m$ is sufficiently large; hence, the algorithm does not terminate in the first iteration, given that $\mathcal{E}_{\text{col}}$ has occurred. Let $U_0, \ldots, U_\kappa$ be the output of the algorithm, where κ is a positive integer. Conditioned on the event $\mathcal{E}_{\text{col}}$, we have $m(U_j) \ge \zeta_{\text{sp}}$ for every $j \le \kappa - 1$.

We prove the claim by induction on κ. For each $j \ge 0$, let $\mathcal{E}_j$ denote the event which is that the algorithm executes iteration j and either terminates with, $m(U_j) \le 8\zeta_{\text{sp}}$, or it produces an approximation preserving subset U_{j+1} such that $m(U_{j+1}) \le \frac{3m(U_j)}{4}$.

Let $\mathcal{E}_{\le j}$ denote the event that $\mathcal{E}_i$ occurs for all $i \in [j]$, that is,

$$\mathcal{E}_{\le j} = \bigcap_{i=1}^{j} \mathcal{E}_i.$$

Equivalently, $\mathcal{E}_{\leq j}$ is the event that the algorithm runs for j iterations, such that $U_0, \ldots, U_j$ is an approximation preserving sequence, and at iteration j either it produces U_{j+1}, an approximation preserving subset of U_j such that $m(U_{j+1}) \leq \frac{3m(U_j)}{4}$, or the algorithm terminates with $m(U_j) \leq 8\zeta_{\text{sp}}$.

The induction hypothesis is that

$$\Pr[\mathcal{E}_{\leq j}] \;\geq\; 1 - \delta_{\leq j},$$

where

$$\delta_i \;=\; \frac{1}{m(U_i)} + \frac{1}{2^{i+5}}, \qquad \text{and} \qquad \delta_{\leq j} \;=\; \sum_{i=0}^{j} \delta_i.$$

We first prove the base case. That is, $\Pr[\mathcal{E}_0] \geq 1 - \delta_0$. Recall that we assume that $m(U_0) = m$ is sufficiently large. Moreover, we assume that event $\mathcal{E}_{\text{col}}$ has occurred and algorithm produced a sequence $U_0, \ldots, U_\kappa$ with $\kappa \geq 1$. So, $\mathcal{E}_0$ occurs if `ApproxSubset` reports an approximate preserving subset U_{j+1}. By Lemma 7, this occurs with probability at least $1 - \delta_0$.

For the inductive step, we mainly show that $\Pr[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}] \geq 1 - \delta_j$ for $j \geq 1$. Let us argue that it is sufficient. Note that $\delta_{\leq j} = \delta_{\leq j-1} + \delta_j$ for $j \geq 1$ and

$$\Pr[\mathcal{E}_{\leq j}] = \Pr[\mathcal{E}_{\leq j-1} \cap \mathcal{E}_j] = \Pr[\mathcal{E}_{\leq j-1}] \cdot \Pr[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}].$$

By the induction hypothesis, $\Pr[\mathcal{E}_{\leq j-1}] \geq 1 - \delta_{\leq j-1}$. We will show that

$$\Pr[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}] \geq 1 - \delta_j.$$

Therefore,

$$\begin{aligned}\Pr\left[\mathcal{E}_{\leq j}\right] &= \Pr\left[\mathcal{E}_{\leq j-1}\right] \cdot \Pr\left[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}\right] \\ &\geq (1 - \delta_{\leq j-1})(1 - \delta_j) \\ &\geq 1 - \delta_{\leq j-1} - \delta_j \\ &= 1 - \delta_{\leq j}.\end{aligned}$$

Consider any j with $1 \leq j \leq \kappa - 1$. Under the conditional space $\mathcal{E}_{\text{col}}$ and along with the fact that we assume $m(U_0) = m$ is sufficiently large, $m(U_j) \geq \zeta_{\text{sp}}$ for each $j \leq \kappa - 1$.

We consider two cases: $m(U_j) \geq 8\zeta_{\text{sp}}$, and $\zeta_{\text{sp}} \leq m(U_j) \leq 8\zeta_{\text{sp}}$, and analyse them individually.

Case 1: $(m(U_j) > 8\zeta_{\text{sp}})$. Recall that we are assuming that the event $\mathcal{E}_{\text{col}}$ has occurred. So, `ApproxSequence` does not terminate after executing `IsCollision` and execute `ApproxSubset`. Moreover it produces an approximate preserving subset U_{j+1} with $m(U_{j+1}) \leq \frac{3m(U_j)}{4}$, that is event $\mathcal{E}_j$ occurs, when `ApproxSubset` reports an approximate preserving subset U_{j+1} with $m(U_{j+1}) \leq \frac{3m(U_j)}{4}$. By Lemma 7 and Line 7 of `ApproxSequence`, this occurs with probability at least $1 - \delta_j$ where,

$$\delta_j = \frac{1}{m(U_j)} + \frac{1}{2^{j+5}}$$

Hence,

$$\Pr[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}] \geq 1 - \delta_j^{(1)},$$

Case 2: $(\zeta_{\mathrm{sp}} \leq m(U_j) \leq 8\zeta_{\mathrm{sp}})$. In this case, we show that

$$\Pr[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}] \geq 1 - \delta_j$$

regardless of whether `ApproxSequence` terminates after executing `IsCollision` or proceeds to execute `ApproxSubset`.

In the former case, the claim follows directly from the definition of $\mathcal{E}_j$ together with the fact that $m(U_j) \leq 8\zeta_{\mathrm{sp}}$. In the latter case, `ApproxSequence` produces an approximate preserving subset U_{j+1} of U_j provided that `ApproxSubset` returns an approximate preserving subset U_{j+1}. By Lemma 7, this occurs with probability at least $1 - \delta_j$, where

$$\delta_j \leq \frac{1}{m(U_j)} + \frac{1}{2^{j+5}}.$$

Hence,

$$\Pr[\mathcal{E}_j \mid \mathcal{E}_{\leq j-1}] \geq 1 - \delta_j.$$

We are done with the proof of the inductive step. We summarize the above discussion as follows. $\Pr[\mathcal{E}_{\mathrm{col}}] \geq 15/16$. When the event $\mathcal{E}_{\mathrm{col}}$ occurs, $\Pr[\mathcal{E}_j \mid \mathcal{E}_{j-1}] \geq 1 - \delta_j$ for all $j \geq 0$. Moreover, $\Pr[\mathcal{E}_{\leq j}] \geq 1 - \delta_{\leq j}$ for all $j \geq 0$. Recall that the algorithm outputs a sequence $U_0, \ldots, U_{\kappa-1}$. The events $\mathcal{E}_{\mathrm{col}}$ and $\mathcal{E}_{\leq \kappa-1}$ occur with probability at least

$$\Pr[\mathcal{E}_{\mathrm{col}}] - \delta_{\leq \kappa-1}.$$

Hence, the following facts hold with the same probability:

1. $U_0, \ldots, U_\kappa$ is an approximate preserving sequence, that is,
$$\frac{1}{32} \leq \frac{m(U_i)}{m(U_{i-1})} \leq \frac{3}{4}, \text{ where } i \in [\kappa].$$
2. $m(U_j) \geq \zeta_{\mathrm{sp}}$ for each $j \leq \kappa - 1$, and $m(U_\kappa) \leq 8\zeta_{\mathrm{sp}}$.

Note that `ApproxSequence` produces an approximation-preserving sequence $U_1, U_2, \cdots, U_\kappa$ such that $m(U_\kappa) \leq 8\zeta_{\mathrm{sp}}$ whenever both of the above conditions hold. This occurs with probability at least $1/5$, provided $\delta_{\kappa-1} \leq 1/10$, since $\Pr[\mathcal{E}_{\mathrm{col}}] \geq 15/16$.

We complete the proof of correctness of `ApproxSequence` by showing that $\delta_{\kappa-1} \leq 1/10$.

$$\delta_{\leq \kappa-1} = \sum_{i=0}^{\kappa-1} \frac{1}{m(U_j)} + \frac{1}{2^{j+5}} \leq \sum_{i=0}^{\infty} \frac{1}{2^{j+4}} + \sum_{i=0}^{\kappa-1} \frac{1}{m(U_j)} \leq \frac{1}{16} + \sum_{i=0}^{\kappa-1} \frac{3^{\kappa-1-i}}{4^{\kappa-1-i}\zeta_{\mathrm{sp}}}.$$

The last inequation holds since $m(U_{\kappa-1}) \geq \zeta_{\mathrm{sp}}$ and $m(U_i) \leq \frac{3m(U_{i-1})}{4}$ for all $i \in [\kappa - 1]$. We obtain the bound of $\delta_{\leq \kappa-1}$ as below:

$$\delta_{\leq \kappa-1} = \frac{1}{16} + \frac{1}{\zeta_{\mathrm{sp}}} \sum_{i=0}^{\kappa-1} \frac{3^{\kappa-1-i}}{4^{\kappa-1-i}} = \frac{1}{16} + \frac{1}{\zeta_{\mathrm{sp}}} \sum_{i=0}^{\kappa-1} \frac{3^i}{4^i} \leq \frac{1}{16} + \frac{2}{\zeta_{\mathrm{sp}}} \leq \frac{1}{10}.$$

Here we have used the fact that ζ_{sp} is a large constant.

Query Complexity. From Lemma 2 and 11, the number of `ISC` queries made by `IsCollision` (for a constant threshold $\tau = \zeta_{\mathrm{sp}}$) and `ApproxSubset` over κ iterations is,

$$O\left(\sum_{j=1}^{\kappa} \log(2^{\kappa+5})\right) = O\left(\kappa^2\right) \qquad \blacktriangleleft$$

Note that the number of iterations and the query complexity of `ApproxSequence` (as mentioned in the above lemma) are random variables. We will argue their expectation in the following lemma. Moreover, when we discuss the full algorithm in Section 5 (that uses `ApproxSequence` as a subroutine), we show that the query complexity of the full algorithm is $O(\kappa^2 \cdot \log \kappa)$, where κ is the number of iterations of `ApproxSequence`. The following lemma will be useful in establishing the final query complexity in Section 5.

▶ **Lemma 13.** $\mathbb{E}[\kappa] = O(\log m)$, $\mathbb{E}[\kappa^2] = O(\log^2 m)$, *and* $\mathbb{E}[\kappa^2 \log \kappa] = O(\log^2 m \cdot \log\log m)$.

From Lemma 12 and Lemma 13, Lemma 6 follows. In the rest of the section, we prove Lemma 13 and Lemma 6.

Proof of Lemma 13. Here, we prove that

$$\mathbb{E}\left[\kappa^2 \log \kappa\right] = O(\log^2 m \log\log m)$$

The remaining bounds follow similarly.

We want to analyse $\mathbb{E}[\kappa^2 \log \kappa]$ where the expectation is over the distribution of κ for a run of the algorithm. To analyse this, we consider two different cases. For the first case, let t^* denote the time when the algorithm generates a subgraph with less than or equal to ζ_{sp} edges. Formally, we define t^* as the time t such that $m(U_t) \leq \zeta_{\text{sp}}$, and $m(U_{t-1}) > \zeta_{\text{sp}}$. If this is not the case, i.e. when $m(U_t) > \zeta_{\text{sp}}$ holds throughout the algorithm, we show that $\kappa = O(\log m)$.

First, we bound the expectation of $m(U_t)$ for some $t \in [\kappa]$. By Lemma 7, we have

$$\mathbb{E}\left[m(U_t) \mid m(U_{t-1})\right] \leq \frac{m(U_{t-1})}{2}.$$

By a inductive argument, and the fact that $m(U_0) = m$, we have

$$\mathbb{E}[m(U_t)] = \mathbb{E}[\mathbb{E}[m(U_t) \;\mid\; m(U_{t-1})]] \leq \frac{\mathbb{E}[m(U_{t-1})]}{2} \leq \ldots \leq \frac{m}{2^t} \tag{3}$$

Now, when $m(U_j) > \zeta_{\text{sp}}$ holds throughout the algorithm, we have

$$\zeta_{\text{sp}} < \mathbb{E}\left[m(U_\kappa)\right] \leq \frac{m}{2^\kappa} \implies \kappa = O(\log m)$$

Hence, we now focus on the case when there exists a $t^* \leq \kappa$. Note that t^* is unique for a run of the algorithm. Once, the algorithm reaches t^*, the probability of the algorithm terminating depends on the calls to `IsCollision` succeeding. We start with analysing the conditional expectation of $\kappa^2 \log \kappa$ given t^*.

$$\begin{aligned}
\mathbb{E}\left[\kappa^2 \log \kappa \mid t^*\right] &= \sum_{j=1}^{\infty} j^2 \log j \Pr\left[\kappa = j \mid t^*\right] \\
&= \sum_{j=1}^{t^*-1} j^2 \log j \Pr\left[\kappa = j \mid t^*\right] + \sum_{j=t^*}^{\infty} j^2 \log j \Pr\left[\kappa = j \mid t^*\right] \\
&= \sum_{j=t^*}^{\infty} j^2 \log j \Pr\left[\kappa = j \mid t^*\right] && \text{(As } t^* \leq \kappa)
\end{aligned}$$

$$\leq \sum_{j=t^*}^{\infty} j^2 \log j \Pr\left[\kappa \geq j \mid t^*\right]$$
$$\leq \sum_{j=t^*}^{\infty} \frac{j^2 \log j}{2^{j+5}}$$
$$= O\left({t^*}^2 \log t^*\right) \tag{4}$$

Now, we analyse the distribution of t^*. We use (3) to obtain a bound on the probabilities of different values of t^*.

$$\begin{aligned}\Pr\left[t^* = t\right] \leq \Pr\left[t^* \geq t\right] \leq \Pr\left[m(U_t) > \zeta_{\text{sp}}\right]& \\ \leq \frac{\mathbb{E}[m(U_t)]}{\zeta_{\text{sp}}} & \qquad \text{Markov's Inequality (Lemma 17)} \\ \leq \frac{m}{2^t \zeta_{\text{sp}}} & \qquad \text{Equation (3)} \end{aligned} \tag{5}$$

Now, we bound the expectation of $\kappa^2 \log \kappa$.

$$\begin{aligned}\mathbb{E}[\kappa^2 \log \kappa] &= \mathbb{E}\left[\mathbb{E}\left[\kappa^2 \log \kappa \mid t^*\right]\right] \\ &= \mathbb{E}[{t^*}^2 \log t^*] \qquad \text{(Equation (4))} \\ &= \sum_{j=1}^{\infty} j^2 \log j \Pr[t^* = j] \\ &= \sum_{j=1}^{2\log m} j^2 \log j \Pr[t^* = j] + \sum_{j=2\log m+1}^{\infty} j^2 \log j \Pr[t^* = j] \\ &\leq 8 \log^2 m \log\log m \sum_{j=1}^{2\log m} \Pr[t^* = j] + \sum_{2\log m}^{\infty} \frac{m j^2 \log j}{2^j \zeta_{\text{sp}}} \\ &= O\left(\log^2 m \log\log m\right)\end{aligned}$$

which proves the desired result. ◀

Lemma 6 follows from the algorithm `ApproxSequence` (Algorithm 2) along with Lemma 12 and Lemma 13.

5 Full algorithm

In this section, we present the full algorithm for estimating the number of edges using `ISC` queries and prove our main result stated below.

▶ **Theorem 1** (Main result). *There exists a randomized algorithm that, given access to a graph G through `ISC` queries and an approximation parameter $\varepsilon \in (0, 1/3)$, outputs an estimate $\widehat{m}$ of the number of edges m in G such that, with probability at least $2/3$,*

$$(1-\varepsilon)m \ \leq\ \widehat{m} \ \leq\ (1+\varepsilon)m.$$

The algorithm uses $O(\varepsilon^{-2} \log^2 m \cdot \log\log m)$ `ISC` queries in expectation.

`EdgeEstimation` (Algorithm 4) begins by calling `ApproxSequence` which outputs an approximation preserving sequence $U_1, \ldots, U_\kappa$, where each $p_i = m(U_i)/m(U_{i-1})$ is bounded below by a constant. This ensures that each ratio can be estimated efficiently. To estimate each p_i, we use the subroutine `MulEstimation` which returns an $(1 \pm \varepsilon)$-approximation of p_i. The procedure `MulEstimation` is described in Algorithm 3 and its guarantee is established in Lemma 14.

Algorithm 3 `MulEstimation`$(U', U, \varepsilon, \delta)$.

Require: $U' \subseteq U$ and `ISC` access to the subset $U \subseteq V$.
Ensure: $\widehat{p}(U')$

1: Uniformly and Independently sample $O\left(\log(\delta^{-1})/\varepsilon^2\right)$ random edges from U and let S be the multi-set of sampled edges.
2: **return** $\widehat{p}(U') = \frac{1}{|S|} \sum_{e \in S} \mathbf{1}_{e \in U'}$.

Algorithm 4 `EdgeEstimation`(V, ε).

Require: `ISC` access to any subset $U \subseteq V$.
Ensure: An estimate of $\widehat{m}$

1: $U_1, U_2, \cdots, U_\kappa \leftarrow$ `ApproxSequence`(V)
2: $\widehat{m}_b \leftarrow$ `BirthdayEstimate`$(U_\kappa, 8\zeta_{\text{sp}}, \frac{\varepsilon}{3})$
3: Initialize $\widehat{p}_i \leftarrow 1$, for all $i \in [\kappa]$
4: **for** $i = 1$ to κ **do**
5: $\quad\widehat{p}_i \leftarrow$ `MulEstimation`$(U_i, U_{i-1}, \frac{\varepsilon}{200\sqrt{\kappa}}, \frac{1}{16\kappa})$
6: **end for**
7: $\widehat{m} \leftarrow \widehat{m}_b / \prod_{i=1}^{\kappa} \widehat{p}_i$
8: **return** $\widehat{m}$

▶ **Lemma 14** (Correctness of `MulEstimation`). *For subsets $U' \subseteq U \subseteq V$ and $\varepsilon > 0$, If $p(U') = \frac{m(U')}{m(U)} \geq \frac{1}{32}$, then `MulEstimation` outputs $\widehat{p}(U') = (1+x)p(U')$, where $\mathbb{E}[x] = 0$ and $x \in (-\varepsilon, \varepsilon)$ with probability at least $1-\delta$. The number of `ISC` queries made is $O\left(\log(\delta^{-1})/\varepsilon^2\right)$.*

Proof. Let X_j be the indicator random variable such that,

$$X_j = \begin{cases} 1 & \text{If the } j^{th} \text{ sampled edge is in } U' \\ 0 & \text{Otherwise} \end{cases}$$

Define number of edges sampled that belong to U' is $X = \sum_{j \in [|S|]} X_j$, where S is the multi-set of sampled edges (Line 2) and the expected size of X is

$$\mathbb{E}[X] = \sum_{j \in [|S|]} \mathbb{E}[X_j] = |S| \cdot \frac{m(U')}{m(U)} = |S| \cdot p(U') \geq \frac{|S|}{32}$$

Therefore,

$$\mathbb{E}[x] = \mathbb{E}\left[\frac{\widehat{p}(U')}{p(U')}\right] - 1 = \frac{\mathbb{E}[X]}{|S|\, p(U')} - 1 = 0$$

By the Chernoff Bound (see Lemma 19), we have

$$\Pr[x \in (-\varepsilon, \varepsilon)] = \Pr\left[\widehat{p}(U') \in (1 \pm \varepsilon)p(U')\right] = \Pr\left[X \in (1 \pm \varepsilon)\, \mathbb{E}[X]\right] \geq 1 - 2\exp\left(-\frac{\varepsilon^2 |S|}{96}\right) \geq 1 - \delta,$$

where the last step uses $|S| = O\left(\log(\delta^{-1})/\varepsilon^2\right)$. ◀

Now, we prove the guarantees of `EdgeEstimation` In particular, we prove the correctness of `EdgeEstimation` and its query complexity in Lemma 16 and Lemma 15, respectively.

▶ **Lemma 15.** *The expected query complexity of `EdgeEstimation` is $O(\log^2 m \cdot \log\log m)$.*

Proof. From Lemma 14, the number of `ISC` queries made by `MulEstimation` in j-th iteration is

$$O\left(\frac{\log(16\kappa)}{\varepsilon'^2}\right) \text{ where } \varepsilon' = \frac{\varepsilon}{200\sqrt{\kappa}}$$

Summing over κ iterations, we get

$$O\left(\sum_{j=1}^{\kappa} \frac{\log(\kappa)}{\varepsilon'^2}\right) = O\left(\kappa^2 \frac{\log\kappa}{\varepsilon^2}\right)$$

From Lemma 12, the number of `ISC` queries made by `ApproxSequence` is $O(\kappa^2)$. Finally, from Lemma 3, the number of `ISC` queries made by `BirthdayEstimate` is $O\left(\sqrt{8\zeta_{\text{sp}}}/\varepsilon\right) = O(1/\varepsilon)$. Combining all contributions, the total number of `ISC` queries made by `EdgeEstimation` is dominated by `MulEstimation`, and hence is, $O(\kappa^2 \log\kappa/\varepsilon^2)$. From Lemma 13,

$$\mathbb{E}[\kappa^2 \log\kappa] = O(\log^2 m \log\log m)$$

which bounds the expected query complexity. ◀

▶ **Lemma 16.** *In `EdgeEstimation`, the estimate $\widehat{m}$ is an $(1 \pm \varepsilon)$-approximation to m with probability at least $3/4$.*

Proof. Let $\mathcal{E}_1 = \mathcal{E}_{\text{sub}} \cap \mathcal{E}_{\text{bir}}$, where $\mathcal{E}_{\text{sub}}$ be the event that `ApproxSequence` outputs an approximation-preserving sequence $U_1, U_2, \ldots, U_\kappa$ with $m(U_\kappa) \leq 8\zeta_{\text{sp}}$. From Lemma 12 we have $\Pr[\mathcal{E}_{\text{sub}}] \geq 4/5$.

$\mathcal{E}_{\text{bir}}$ be the event that `BirthdayEstimate` outputs an $(1 \pm \varepsilon/3)$ approximation of $m(U_\kappa)$ from Lemma 3. Also, from Lemma 3, we have $\Pr[\mathcal{E}_{\text{bir}} \mid \mathcal{E}_{\text{sub}}] \geq 15/16$. Therefore,

$$\Pr[\mathcal{E}_1] = \Pr[\mathcal{E}_{\text{sub}}] \Pr[\mathcal{E}_{\text{bir}} \mid \mathcal{E}_{\text{sub}}] \geq 1 - \left(\frac{1}{5} + \frac{1}{16}\right) \geq \frac{59}{80}$$

Let $\mathcal{E}_2$ be the event that `MulEstimation`$(U_i, U_{i-1}, \varepsilon', 1/16\kappa)$ outputs $\widehat{p_i} = (1 \pm x_i/4)p_i$ such that $\mathbb{E}[x_i] = 0$ and $x_i \in (-4\varepsilon', 4\varepsilon')$ for all $i \in [\kappa]$, where $\varepsilon' = \frac{\varepsilon}{200\sqrt{\kappa}}$. Therefore, using Lemma 14 and applying the union bound over all $i \in [\kappa]$,

$$\Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - \sum_{j=1}^{\kappa} \frac{1}{16\kappa} \geq \frac{15}{16}.$$

Let $\mathcal{E}_3$ be the event that $|\sum_{i=1}^{\kappa} x_i| \leq \varepsilon/9$. By Hoeffding's bound (Lemma 20),

$$\begin{aligned}\Pr[\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2] &= 1 - \Pr\left[\left|\sum_{i=1}^{\kappa} x_i\right| > \frac{\varepsilon}{9} \middle| \mathcal{E}_1 \cap \mathcal{E}_2\right] \\ &\geq 1 - 2\exp\left(-\frac{2\varepsilon^2}{81 \sum_{i=1}^{\kappa} \left(\frac{2\varepsilon}{50\sqrt{\kappa}}\right)^2}\right) \geq 1 - \frac{1}{e^{12}}.\end{aligned} \tag{6}$$

Hence,

$$\Pr[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3] = \Pr[\mathcal{E}_1] \cdot \Pr[\mathcal{E}_2|\mathcal{E}_1] \cdot \Pr[\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2] \geq 1 - \left(\frac{21}{80} + \frac{1}{16} + \frac{1}{e^{12}}\right) \geq \frac{2}{3}$$

In what follows, we assume that all of the events $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ occur. Since, $\Pr[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3] \geq 2/3$, it is enough to argue that `EdgeEstimation` returns an $(1 \pm \varepsilon)$-approximation to m when all $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ holds.

Combining the output of `MulEstimation` over the κ iterations, we get,

$$\prod_{i=1}^{\kappa} \widehat{p}_i = \prod_{i=1}^{\kappa} p_i \cdot \prod_{i=1}^{\kappa} \left(1 + \frac{x_i}{4}\right)$$

Using the fact that for each $x_i \in (0, 1)$, it can be shown that

$$1 - \frac{\sum_{i=1}^{\kappa} x_i}{4} \leq \prod_{i=1}^{\kappa} \left(1 + \frac{x_i}{4}\right) \leq 1 + \sum_{i=1}^{\kappa} x_i$$

Since event $\mathcal{E}_3$ occurs,

$$\prod_{i=1}^{\kappa} p_i \cdot \left(1 - \frac{\varepsilon}{36}\right) \leq \prod_{i=1}^{\kappa} \widehat{p}_i \leq \prod_{i=1}^{\kappa} p_i \cdot \left(1 + \frac{\varepsilon}{9}\right)$$

Using the fact that for any $\varepsilon \in (0, 1/3)$, we have $1 - 3\varepsilon \leq (1 + \varepsilon)^{-1}$ and $(1 - \varepsilon)^{-1} \leq 1 + 3\varepsilon$. We get,

$$\left(1 - \frac{\varepsilon}{3}\right) \cdot \frac{1}{\prod_{i=1}^{\kappa} p_i} \leq \frac{1}{\prod_{i=1}^{\kappa} \widehat{p}_i} \leq \frac{1}{\prod_{i=1}^{\kappa} p_i} \cdot \left(1 + \frac{\varepsilon}{12}\right) \tag{7}$$

Since $\mathcal{E}_1$ occurs, i.e., $\mathcal{E}_{\mathrm{bir}}$ occurs, $\widehat{m}_b$ satisfies the following:

$$\left(1 - \frac{\varepsilon}{3}\right) \cdot m(U_\kappa) \leq \widehat{m}_b \leq \left(1 + \frac{\varepsilon}{3}\right) \cdot m(U_\kappa) \tag{8}$$

Combining (7) and (8), we get,

$$\left(1 - \frac{\varepsilon}{3}\right)^2 \cdot \frac{m(U_\kappa)}{\prod_{i=1}^{\kappa} p_i} \leq \widehat{m} = \frac{\widehat{m}_b}{\prod_{i=1}^{\kappa} \widehat{p}_i} \leq \left(1 + \frac{\varepsilon}{12}\right)\left(1 + \frac{\varepsilon}{3}\right) \cdot \frac{m(U_\kappa)}{\prod_{i=1}^{\kappa} p_i}$$

Since, $p_i = \frac{m(U_i)}{m(U_{i-1})}$ for each $i \in [\kappa]$ and $U_0 = V$, $\frac{m(U_\kappa)}{\prod_{i=1}^{\kappa} p_i} = m(U_0) = m$. So, simplifying the above expression, we have

$$(1 - \varepsilon)\, m \leq \widehat{m} \leq (1 + \varepsilon)\, m. \qquad \blacktriangleleft$$

References

1 Tomer Adar, Yahel Hotam, and Amit Levi. When Local and Non-Local Meet: Quadratic Improvement for Edge Estimation with Independent Set Queries, 2026. arXiv:2601.21457. `doi:10.48550/arXiv.2601.21457`.

2 Raghavendra Addanki, Andrew McGregor, and Cameron Musco. Non-Adaptive Edge Counting and Sampling via Bipartite Independent Set Queries. In *Proceedings of the 30th Annual European Symposium on Algorithms, ESA*, pages 2:1–2:16, 2022. `doi:10.4230/LIPIcs.ESA.2022.2`.

3 Matthew Aldridge, Oliver Johnson, and Jonathan Scarlett. Group Testing: An Information Theory Perspective. *Foundations and Trends in Communications and Information Theory*, 15(3-4):196–392, 2019. `doi:10.1561/0100000099`.

4 Maryam Aliakbarpour, Amartya Shankha Biswas, Themis Gouleakis, John Peebles, Ronitt Rubinfeld, and Anak Yodpinyanee. Sublinear-Time Algorithms for Counting Star Subgraphs via Edge Sampling. *Algorithmica*, 80(2):668–697, 2018. `doi:10.1007/S00453-017-0287-3`.

5 Sepehr Assadi, Deeparnab Chakrabarty, and Sanjeev Khanna. Graph Connectivity and Single Element Recovery via Linear and OR Queries. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA)*, pages 7:1–7:19, 2021. `doi:10.4230/LIPIcs.ESA.2021.7`.

6 Paul Beame, Sariel Har-Peled, Sivaramakrishnan Natarajan Ramamoorthy, Cyrus Rashtchian, and Makrand Sinha. Edge Estimation with Independent Set Oracles. *ACM Transactions on Algorithms*, 16(4):52:1–52:27, 2020. preliminary version of this paper appeared as an extended abstract in the proceedings of ITCS 2018. `doi:10.1145/3404867`.

7 Lorenzo Beretta, Deeparnab Chakrabarty, and C. Seshadhri. Faster Estimation of the Average Degree of a Graph Using Random Edges and Structural Queries. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 939–971, 2026. `doi:10.1137/1.9781611978971.38`.

8 Anup Bhattacharya, Arijit Bishnu, Arijit Ghosh, and Gopinath Mishra. Faster Counting and Sampling Algorithms using Colorful Decision Oracle. *ACM Transactions on Computation Theory*, 16(2):12:1–12:19, 2024. A preliminary version of this paper appeared as an extended abstract in the proceedings of STACS 2022. `doi:10.1145/3657605`.

9 Rishiraj Bhattacharyya and Sourav Chakraborty. Property Testing of Joint Distributions using Conditional Samples. *ACM Transactions on Computation Theory*, 10(4):16:1–16:20, 2018. `doi:10.1145/3241377`.

10 Rishiraj Bhattacharyya, Sourav Chakraborty, Yash Pote, Uddalok Sarkar, and Sayantan Sen. Testing Self-Reducible Samplers. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7952–7960, 2024. `doi:10.1609/AAAI.V38I8.28632`.

11 Arijit Bishnu, Debarshi Chanda, Buddha Dev Das, Arijit Ghosh, and Gopinath Mishra. Optimal non-adaptive algorithm for edge estimation. *CoRR*, abs/2512.11994, 2025. `doi:10.48550/arXiv.2512.11994`.

12 Arijit Bishnu, Debarshi Chanda, and Gopinath Mishra. Towards tight bounds for estimating degree distribution in streaming and query models. *CoRR*, abs/2507.21784, 2025. `doi:10.48550/arXiv.2507.21784`.

13 Arijit Bishnu, Arijit Ghosh, and Gopinath Mishra. On the Complexity of Triangle Counting Using Emptiness Queries. In *Proceedings of the 27th International Conference on Randomization and Computation (RANDOM)*, volume 275, pages 48:1–48:22, 2023. `doi:10.4230/LIPIcs.APPROX/RANDOM.2023.48`.

14 Clément L. Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. Random Restrictions of High Dimensional Distributions and Uniformity Testing with Subcube Conditioning. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 321–336, 2021. `doi:10.1137/1.9781611976465.21`.

15 Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing equivalence between distributions using conditional samples. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1174–1192, 2014. `doi:10.1137/1.9781611973402.87`.

16 Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing Probability Distributions using Conditional Samples. *SIAM Journal on Computing*, 44(3):540–616, 2015. An extended abstract of this work appeared in SODA 2014. `doi:10.1137/130945508`.

17 Diptarka Chakraborty, Sourav Chakraborty, Gunjan Kumar, and Kuldeep S. Meel. Approximate Model Counting: Is SAT Oracle More Powerful Than NP Oracle? In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 261, pages 123:1–123:17, 2023. `doi:10.4230/LIPIcs.ICALP.2023.123`.

18 Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 561–580, 2013. `doi:10.1145/2422436.2422497`.

19 Debarshi Chanda. Improved Bounds with a Simple Algorithm for Edge Estimation for Graphs of Unknown Size. *CoRR*, abs/2511.03650, 2025. `doi:10.48550/arXiv.2511.03650`.

20 Debarshi Chanda, Buddha Dev Das, Arijit Ghosh, and Gopinath Mishra. From decision to random certificates: Exponential separation for edge estimation with independent set queries, 2026. `arXiv:2607.07483`.

21 Xi Chen, Amit Levi, and Erik Waingarten. Nearly Optimal Edge Estimation with Independent Set Queries. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2916–2935, 2020. `doi:10.1137/1.9781611975994.177`.

22 Fan Chung and Linyuan Lu. Concentration Inequalities and Martingale Inequalities: A Survey. *Internet Mathematics*, 3(1):79–127, 2006.

23 Holger Dell, John Lapinskas, and Kitty Meeks. Approximately Counting and Sampling Small Witnesses Using a Colorful Decision Oracle. *SIAM Journal on Computing*, 51(4):849–899, 2022. A preliminary version of this paper appeared as an extended abstract in the proceedings of SODA 2020. `doi:10.1137/19M130604X`.

24 Holger Dell, John Lapinskas, and Kitty Meeks. Nearly Optimal Independence Oracle Algorithms for Edge Estimation in Hypergraphs. In *Proceedings of the 51st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 54:1–54:17, 2024. `doi:10.4230/LIPIcs.ICALP.2024.54`.

25 Robert Dorfman. The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

26 Talya Eden, Dana Ron, and C. Seshadhri. Sublinear Time Estimation of Degree Distribution Moments: The Arboricity Connection. *SIAM Journal on Discrete Mathematics*, 33(4):2267–2285, 2019. `doi:10.1137/17M1159014`.

27 Talya Eden and Will Rosenbaum. Lower Bounds for Approximating Graph Parameters via Communication Complexity. In *Proceedings of the 21st International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 11:1–11:18, 2018. `doi:10.4230/LIPIcs.APPROX-RANDOM.2018.11`.

28 Uriel Feige. On Sums of Independent Random Variables with Unbounded Variance and Estimating the Average Degree in a Graph. *SIAM Journal on Computing*, 35(4):964–984, 2006. An extended abstract of this paper appeared in the proceedings of STOC 2004. `doi:10.1137/S0097539704447304`.

29 Oded Goldreich and Dana Ron. Approximating Average Parameters of Graphs. *Random Structures & Algorithms*, 32(4):473–493, 2008. An extended abstract of this paper appeared in the proceedings of RANDOM 2006. `doi:10.1002/RSA.20203`.

30 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, Cambridge, 2005. `doi:10.1017/CBO9780511813603`.

31 Dana Ron and Gilad Tsur. The Power of an Example: Hidden Set Size Approximation Using Group Queries and Conditional Sampling. *ACM Transactions on Computation Theory*, 8(4), 2016. `doi:10.1145/2930657`.

32 Larry J. Stockmeyer. The Complexity of Approximate Counting (Preliminary Version). In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 118–126, 1983. `doi:10.1145/800061.808740`.

33 Larry J. Stockmeyer. On Approximation Algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985. `doi:10.1137/0214060`.

34 Jakub Tětek and Mikkel Thorup. Edge Sampling and Graph Parameter Estimation via Vertex Neighborhood Accesses. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1116–1129, 2022.

A Concentration Inequalities

We use the following concentration bounds.

▶ **Lemma 17** (Markkov's inequlaity, see [30]). *Let X be a random variable that assumes only nonnegative values. Then, for all $a > 0$*

$$\Pr\left[X \geq a\right] \leq \frac{\mathbb{E}[X]}{a}.$$

▶ **Lemma 18** (Chebyshev's inequality, see [30]). *Let X be a random variable and $a > 0$, Then*

$$\Pr[|X - \mathbb{E}[X]| > a] \leq \frac{\mathrm{Var}(X)}{a^2}.$$

▶ **Lemma 19** (Multiplicative Chernoff bound, see [30]). *Let $X_1, X_2, ..., X_t$ be i.i.d. random variables where $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ for all $i \in [t]$, and $X = \sum_{i \in [t]} X_i$. Then, we have*

$$\Pr\left[X \leq (1-\varepsilon)\,\mathbb{E}\left[X\right]\right] \leq \exp\left(-\frac{\varepsilon^2\,\mathbb{E}\left[X\right]}{3}\right) \qquad 0 \leq \varepsilon < 1$$

$$\Pr\left[X \geq (1+\varepsilon)\,\mathbb{E}\left[X\right]\right] \leq \exp\left(-\frac{\varepsilon^2\,\mathbb{E}\left[X\right]}{2}\right) \qquad \varepsilon \geq 0$$

▶ **Lemma 20** (Hoeffding's bound, see [30]). *Let $X_1, X_2, \ldots, X_t$ be independent random variables such that for all $i \in [t]$, $\mathbb{E}[X] = \mu$ and $\Pr[a \leq X_i \leq b] = 1$. Then*

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} X_i - \mu\right| \geq \varepsilon\right] \leq 2\exp\left(-\frac{2t\varepsilon^2}{(b-a)^2}\right).$$

▶ **Lemma 21** (See [22]). *Let f be any function of t independent random variables $Y_1, \ldots, Y_t$ and let $X_i = \mathbb{E}[f(Y_1, \ldots, Y_t)|Y_1, \ldots, Y_i]$ for $i \in [t]$ and $X_0 = \mathbb{E}[f(Y_1, \ldots, Y_t)]$. Let $\mathcal{B}$ be the event that there exists $i \in [t]$ such that $|X_i - X_{i-1}| > c_i$ where $c_1, \ldots, c_t$ are some non negative numbers. Let $S = \sum_{i=1}^{t} c_i^2$, then*

$$\Pr\left[|X_t - X_0| > s\right] \leq 2\exp\left(-\frac{s^2}{2S}\right) + \Pr[\mathcal{B}].$$