COMPARING THE STRENGTH OF QUERY TYPES IN PROPERTY TESTING: THE CASE OF TESTING K-COLORABILITY

Ido Ben-Eliezer, Tali Kaufman, Michael Krivelevich, and Dana Ron

Abstract. We study the power of four query models in the context of property testing in general graphs, where our main case study is the problem of testing k-colorability. Two query types, which have been studied extensively in the past, are *pair queries* and *neighbor queries*. The former corresponds to asking whether there is an edge between any particular pair of vertices, and the latter to asking for the i^{th} neighbor of a particular vertex. We show that while for pair queries, testing k-colorability requires a number of queries that is a monotone decreasing function in the average degree d, the query complexity in the case of neighbor queries remains roughly the same for every density and for large values of k. We also consider a combined model that allows both types of queries, and we propose a new, stronger, query model, related to the field of Group Testing. We give upper and lower bounds on the query complexity for one-sided error in all the models, where the bounds are nearly tight for three of the models. In some of the cases our lower bounds extend to two-sided error algorithms.
The problem of testing k-colorability was previously studied in the contexts of dense graphs and of sparse graphs, and in our proofs we unify approaches from those cases, and also provide some new tools and techniques that may be of independent interest.

Keywords. Property Testing, k-colorability

Subject classification. 68Q17, 68Q25, 05C15, 05C85

1. Introduction

Property testing [14, 24] deals with the problem of deciding whether a certain object has a prespecified property P or it is far (i.e., differs significantly) from any object that has P. Namely, the algorithm should *accept* objects that have the property, and should *reject* objects that are far from having the property with respect to some predetermined distance measure, where the algorithm is allowed a small probability of failure. The algorithm is given query access to the object, and it should make the decision after observing only a small part of the object. Thus, the main complexity measure studied in the context of property testing is the *query complexity* of the algorithm, which is normally expected to be *sublinear* in the size of the object, and the question is how this measure varies as a function of various parameters.

In this work we compare the power of different query types in the context of testing graph properties of general graphs (i.e., with arbitrary edge densities). To this end we focus on the problem of testing k-colorability for $k \geq 3$, and study the query complexity of this problem for different query types as a function of the number of vertices and the average degree of the graph.

1.1. The Distance Measure and Query Types Studied in this Paper. When defining models for property testing of graphs there are two issues to consider: the distance measure between graphs (which determines what graphs should be rejected by the testing algorithm) and the types of queries that the algorithm is allowed to make. Since we study graphs of varying edge densities and vertex degrees, we follow [21, 18] and define our distance measure with respect to the total number of edges in the graph. Namely, if n denotes the number of graph vertices and d denotes the average degree, then we say that a graph is ϵ*-far* from being k-colorable for a given $0 \leq \epsilon \leq 1$, if it is necessary to remove more than ϵdn edges so as to obtain a k-colorable graph.[1]

We consider the following types of queries where the first two have been considered in the past and the third is a new query type we introduce.

- Pair queries. These are queries of the form "Is there an edge between the pair of vertices u and v?".
- Neighbor queries. These are queries of the form: "Who is the i^{th} neighbor of vertex v?". If v has less than i neighbors then a special symbol is

[1] Another well studied distance measure is the fraction of edge modification as a function of n^2. This measure is appropriate for dense graphs (i.e., that satisfy $d = \Theta(n)$). In what can be viewed as the other extreme, where all vertices have bounded degree $d_{\max}$ (in particular, $d_{\max} = O(1)$), distance is measured with respect to $d_{\max}n$).

returned, and no assumption is made about the order of the neighbors of a vertex. Therefore, every new query reveals a new random neighbor of a given vertex.

- ◦ Group queries. We propose a new query type that extends pair queries. These queries are of the form "Is there at least one edge between a vertex u and a set of vertices S?".

The study of group queries is partially motivated by the study of Group Testing (see, e.g., [10]), where similar queries are allowed. Problems of group testing can be found in various fields such as Statistics and Biology. As we explain in more detail below, another motivation for studying group queries is that they can serve as a tool for obtaining lower bounds when using the other types of queries, as these queries are essentially stronger than the standard pair and neighbor queries.

In all cases, we also allow our algorithms to perform degree queries. That is, the algorithms may ask for the degree of any vertex of their choice.[2]

In what follows, when we refer to the *pair query model* (respectively, *neighbor query model* and *group query model*), we mean that only pair queries (respectively, neighbor queries and group queries) are allowed, as well as degree queries. When both pair queries and neighbor queries are allowed, then we refer to the resulting model as the *combined model*.

1.2. Related Work on Testing k-Colorability. Testing k-colorability has previously been studied in the pair query model for the case that the graph is dense, that is, $d = \Theta(n)$. For this case, k-colorability is testable using a number of queries that is *independent* of the graph size (and polynomial in k and $1/\epsilon$ [14, 4]).[3]

Testing k-colorability has previously been studied in the neighbor query model for the case that $k = 3$ and the graph has constant maximum degree (that is, $d = O(1)$, and furthermore, the maximum degree $d_{\max}$ is $O(1)$ as

[2]In fact, we shall not need degree queries when using pair queries. Also, allowing degree queries in conjunction with neighbor queries is only done to simplify the presentation, since a degree query can be emulated using $O(\log n)$ neighbor queries. As for group queries, in our context it suffices to use approximate degrees, which can be emulated using group queries. However, in general, in order to get a query model that is as strong as one which allows both pair queries and neighbor queries, we allow degree queries in conjunction with group queries.

[3]Interestingly, the earlier work of Rödl and Duke [22] implicitly implies that k-colorability is testable using a number of queries that is independent of the graph size, but is a tower function of $1/\epsilon$.

well). In this case, Bogdanov et al. [6] proved that is necessary to perform $\Omega(n)$ queries (that is, there is no algorithm with sublinear query complexity).

Testing k-colorability for $k = 2$ (i.e., testing bipartitness) has previously been studied for general graphs in the combined model [18] where it was shown that $\tilde{\Theta}(\min\{\sqrt{n}, n/d\})$ (pair and neighbor) queries are both sufficient and necessary. The proof of the lower bound in [18] implies that if only neighbor queries are allowed then $\Omega(\sqrt{n})$ queries are necessary for every value of d, and if only pair queries are allowed then $\Omega(n/d)$ queries are necessary.[4]

1.3. Our Results. In this work we study the power of the different types of queries when testing k-colorability of general graphs for a fixed $k \geq 3$. In previous work on testing properties of graphs, the pair query model was studied in the case of dense graphs, the neighbor query model was studied in the case of bounded-degree graphs, and for general graphs, the combined model was considered. Here we are interested in understanding how the query complexity of the problem behaves as a function of the edge density (and the number of vertices) when the algorithm is allowed to perform only one type of query, and whether there is a gain when allowing to combine query types. One motivation for this investigation is that what types of queries are allowed intuitively corresponds to how the graph is represented. Thus, allowing both pair queries and neighbor queries (as in the combined model) implicitly assumes that the algorithm has access both to an adjacency matrix representation (supporting pair queries) and to an incidence lists representation (supporting neighbor queries), which is not necessarily the case. Our second motivation is simply complexity theoretic: understanding the strength of each query type separately (and possibly combined) for varying edge densities.

In what follows we say that an algorithm has *one-sided error* if it always accepts graphs that are k-colorable, otherwise it has *two-sided error*. Our results are stated in terms of the dependence on n and d. In all our upper bounds the dependence on both k and $1/\epsilon$ is polynomial. In our lower bounds it is assumed that ϵ is a constant, and unless stated explicitly otherwise (i.e., when k appears in the exponent of the expression for the complexity), k is viewed as a constant as well. With a slight abuse of notation, we write $f = \tilde{O}(g)$ (and similarly, $f = \tilde{\Omega}(g)$) if $f(x) = O(g(x)) \cdot \text{polylog}(n)$ for every x, where n is the number of vertices.

The bounds we present are for the query complexity in the different mod-

[4] In earlier work [16, 15] it was shown that if only neighbor queries are allowed and the distance measure is with respect to $d_{\max}n$ rather than dn, then $\tilde{\Theta}(\sqrt{n})$ are both necessary and sufficient.

els. The running time of our algorithms may be exponential in the number of queries, but the focus of this work is only on the query complexity. We assume for simplicity that the algorithms are given d as input. However, they actually need only a constant factor estimate of d, and this can be obtained by performing $\tilde{O}(\sqrt{n/d})$ degree queries in expectation [12, 17] (which is negligble for our algorithms).

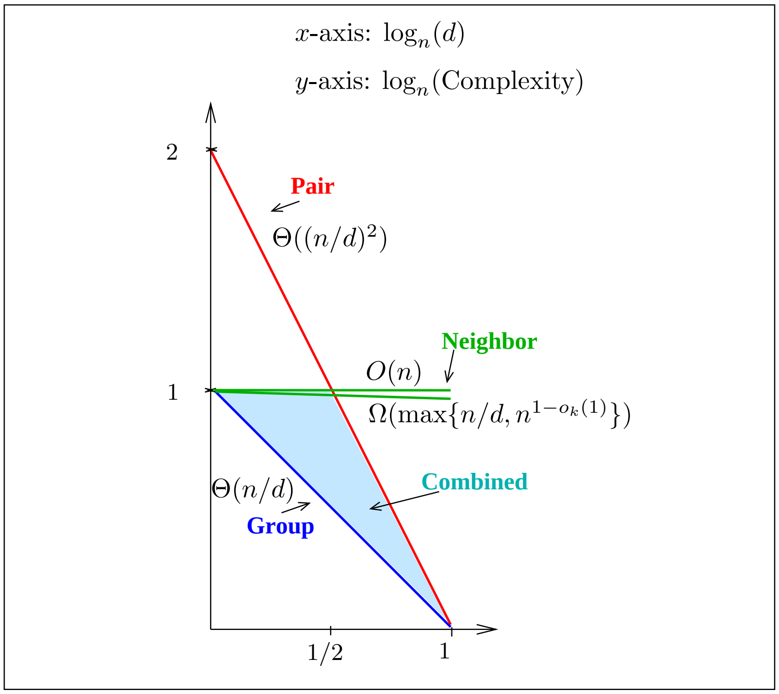

Figure 1.1: A schematic illustration of the query complexity for the different query types (and one-sided error) on a "log-log" scale. For the sake of simplicity we ignore logarithmic factors in the bounds and furthermore, for the neighbor query model we think of k as being large. For the combined model we have that the lower bound on the query complexity coincides with the group query model, and the upper bound coincides with the neighbor query model until $d = \sqrt{n}$ and from that point on it coincides with the pair query model.

Theorem 1.1. *The following holds for testing k-colorability, $k \geq 3$, in the pair query model:*

	Pair	Neighbor	Group	Pair&Neighbor
UB	$\tilde{O}\left(\left(\frac{n}{d}\right)^2\right)$	$O(n)$	$\tilde{O}\left(\frac{n}{d}\right)$	$\min\left\{\tilde{O}\left(\left(\frac{n}{d}\right)^2\right), O(n)\right\}$
LB	$\Omega\left(\left(\frac{n}{d}\right)^2\right)$	$\Omega\left(n^{1-O(1/k)}\right)$ $\Omega\left(n \cdot d^{-O(1/k)}\right)$ if $k \geq 6$	$\Omega\left(\frac{n}{d}\right)$ also for 2-sided error	$\tilde{\Omega}\left(\frac{n}{d}\right)$ also for 2-sided error

Table 1.1: Results for one-sided error testing of k-colorability ("UB" stands for "Upper Bounds" and "LB" stands for "Lower Bounds").

(i) There exists a one-sided error tester that performs $\tilde{O}\left(\left(\frac{n}{d}\right)^2\right)$ *queries.*

(ii) Every one-sided error tester must perform $\Omega\left(\left(\frac{n}{d}\right)^2\right)$ *queries.*

Theorem 1.2. *The following holds for testing k-colorability, $k \geq 3$, in the neighbor query model:*

(i) There exists a one-sided error tester that performs $O(n)$ queries.

(ii) Every tester must perform $\tilde{\Omega}\left(\max\left\{\frac{n}{d}, \sqrt{n}\right\}\right)$ *queries.*

(iii) Every one-sided error tester must perform $\Omega\left(n^{1-\frac{1}{\lceil (k+1)/2 \rceil}}\right)$ *queries.*

(iv) Every one-sided error tester for $k \geq 6$ must perform $\Omega\left(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}}\right)$ *queries.*

Observe that for one-sided error testers in the neighbor query model, as k increases, our lower bound approaches our upper bound.

Theorem 1.3. *The following holds for testing k-colorability, $k \geq 3$, in the group query model:*

(i) There exists a one-sided error tester that performs $\tilde{O}\left(\frac{n}{d}\right)$ *queries.*

(ii) Every tester must perform $\Omega\left(\frac{n}{d}\right)$ *queries.*

By combining Theorems 1.1, 1.2, 1.3 and the fact that neighbor queries can be emulated using a poly-logarithmic number of group queries and degree queries (Claim 2.1), we get the following corollary. The upper bound in the corollary is simply the minimum between the upper bound in Theorem 1.1 and the upper bound in Theorem 1.2, and the lower bound is as in Theorem 1.3 (up to polylogarithmic factors).

Corollary 1.4. *The following holds for testing k-colorability, $k \geq 3$, in the combined query model:*

(i) There exists a one-sided error tester that performs

$\min\left\{\tilde{O}\left(\left(\frac{n}{d}\right)^2\right), O(n)\right\}$ *queries.*

(ii) Every tester must perform $\tilde{\Omega}\left(\frac{n}{d}\right)$ *queries.*

The results are summarized in Table 1.1 and are illustrated in Figure 1.1.

Discussion of our results and conclusions. We next discuss the main phenomena we observe in our study of testing k-colorability in the different query models.

- While the query complexity in the pair query model and the query complexity in the group query model are monotone decreasing functions of d, the query complexity in the neighbor query model remains roughly the same for every value of d (and large values of k).

- When comparing the pair query model to the neighbor query model in more detail we see that the query complexity in the pair query model is higher than in the neighbor query model for $d < \sqrt{n}$, while once d passes $\sqrt{n}$ it becomes lower (and continues decreasing). The extreme case is $d = \Theta(n)$, where in the pair query model the query complexity does not depend on n.[5] This coincides with the intuition that neighbor queries are useful for sparse graphs and pair queries are useful for dense graphs.

[5]In fact, when $d = \Theta(n)$ then many natural properties are testable in the pair query model using a constant number of queries (see e.g. [14, 1, 2]).

- When comparing the group query model to the other models we observe that the query complexity in the former model is never higher (up to poly-logarithmic factors) than in the latter models. This is a general phenomenon since pair queries are a special case of group queries and neighbor queries can be emulated using group queries and degree queries at a poly-logarithmic cost (Claim 2.1).

  We also show that for the case of testing k-colorability and a certain distribution over graphs, the combined model is strictly stronger than the optimum of the pair query and the neighbor query models.

In our proofs we extend and unify methods from previous works, and present several new methods. It turns out that procedures for sampling edges are useful in problems of property testing in general graphs, and here we give more efficient procedures. In the lower bounds section we use directed graphs to represent the so called "knowledge graph" of a testing algorithm. The combination of this representation and balls and bins techniques enables us to prove lower bounds in this model, and in particular in the problem of testing k-colorability.

1.4. Other Related Work. Czumaj et al. [8] show that for the special case of graphs with constant degree that have a certain separating (non-expansion) property, k-colorability (among other properties), is testable using a number of queries that depends only on $1/\epsilon$ (though possibly exponentially).

Testing triangle-freeness (and more generally, subgraph-freeness) of general graphs has been studied in the combined model [3]. The main result in [3] is a lower bound of $\Omega(n^{1/3})$ on the necessary number of queries that holds for every $d < n^{1-\nu(n)}$, where $\nu(n) = o(1)$. This stands in contrast to the complexity of testing triangle-freeness using pair queries when $d = \Theta(n)$ where there is no dependence on n [1].

2. Preliminaries

2.1. Notation and Definitions. Given a graph $G = (V, E)$ on n vertices and average degree d, we say that G is *almost-regular* if the maximal degree is bounded by $10d$ (the constant 10 here is somewhat arbitrary). Denote by $\Gamma(v)$ the set of neighbors of a vertex v. For a set of vertices S, denote by $G[S]$ the induced subgraph of G on the set S, and by $E[S]$ the set of edges of $G[S]$. For a subgraph H of G and a vertex v in H, we let $\Gamma_H(v)$ denote the set of neighbors that v has in H.

Let $[k] \stackrel{\text{def}}{=} \{1, \ldots, k\}$. For a coloring $\varphi : V \to [k]$ (a k-partition of V), we say that an edge $(u, v) \in E$ is *monochromatic* (*violating* with respect to the k-partition), if $\varphi(u) = \varphi(v)$ (both end-points of the edge are in the same subset of the partition). In our setting, a graph is ϵ-far from being k-colorable if for any k-coloring of G we have at least ϵnd violating edges. Unless stated explicitly otherwise, we assume $k \geq 3$.

A *k-critical* graph is a graph that is not $(k-1)$-colorable, but removal of any single edge or vertex will give a $(k-1)$-colorable graph. Every k-critical graph has minimal degree $(k-1)$, and every non-$(k-1)$-colorable graph contains a k-critical graph.

For every execution of a tester, the *knowledge graph* is the graph formed by answers to the queries the tester made. The knowledge graph contains both edges and non-edges (for example, we get a non-edge by a negative answer to a pair query). Clearly, if a one-sided error algorithm decides that a given graph is ϵ-far from being k-colorable, the subgraph formed by the edges of its knowledge graph must contain a $(k+1)$-critical graph.

In what follows, whenever we say "with high probability" we mean with probability $1 - o(1)$. When we apply Chernoff bounds we use either Theorem A.4 or Corollary A.7 from [5].

2.2. Simple emulation of queries. Recall that throughout this work we allow degree queries in our models. A degree query can clearly be emulated by a logarithmic number of neighbor queries.

We next prove two simple claims that demonstrate the power of group queries. In all that follows, when we refer to a *random* neighbor query we mean a query concerning a vertex v that returns a uniformly selected neighbor of v.

CLAIM 2.1. *A pair query can be emulated using a single group query, while a random neighbor query can be emulated using a logarithmic number of group queries. Moreover, every sequence of neighbor queries can be emulated by group queries with a logarithmic overhead.*

PROOF. Clearly, a pair query is a special case of a group query, where the set we query for has a single element. Next we show how to emulate a random neighbor query. For a given vertex v, in order to emulate a random neighbor query for v we need to show how to find a random neighbor of v. This can be done in the following manner: Randomly order the vertices of G as the leaves of a full binary tree. For every internal node y denote by $S(y)$ the set of vertices that correspond to the leaves of the subtree rooted at y, and denote by $C(y)$ the two children of y. Now, apply the following procedure, starting from the root: Given a node y, perform a group query between v and $S(y')$ for every

$y' \in C(y)$. If the two queries return negative answers then v has no neighbors. If exactly one of the queries returns a positive answer, apply the procedure recursively on the node that returned the positive answer. Otherwise, choose one of the children randomly and apply the procedure recursively on it. The procedure ends when it finds a single leaf, and since the vertices were ordered randomly in the leaves, this procedure gives a random neighbor in a logarithmic number of queries.

We conclude by observing that every sequence of q neighbor queries can be emulated by $O(q \log n)$ neighbor queries. Indeed, every new neighbor query asks for a new random neighbor of a given vertex. Given a vertex v, let $T(v)$ be the set of neighbors of v that were found in previous queries. By performing a group query on $V \setminus T(v)$ we can check whether v has a neighbor that was not found yet, and by applying a recursive procedure as above, we can find a random neighbor among the vertices in $V \setminus T(v)$, as needed. $\square$

Next we use a similar idea to show how to find efficiently all the edges of an induced subgraph using group queries.

CLAIM 2.2. *It is possible to find all the edges of an arbitrary induced subgraph of order n' using $\tilde{O}(n' + m')$ group queries where m' is the number of edges in the induced subgraph.*

PROOF. Let U be a set of vertices of size n'. For every vertex $v \in U$, we find all its neighbors in U by the following procedure: Order all the vertices of U as the leaves of a full binary tree. For every internal node y denote by $S(y)$ the set of vertices corresponding to leaves in the subtree rooted at y. Our aim is to find all the leaves that correspond to neighbors of v. Starting from the root, if the node is a leaf, we perform a group query (that is, a pair query) to check if the corresponding vertex is a neighbor of v. Otherwise, perform a group query between v and $S(y)$. If the answer is 'false', return nothing. Otherwise, apply the same procedure recursively on the two children of y, and return the union of their results. It is not hard to verify that this procedure returns all the neighbors of v in U, and for every neighbor we perform a logarithmic number of group queries to find it. Thus, the total number of queries is $\tilde{O}(n' + m')$, as desired. $\square$

3. Upper Bounds

In this section we establish the upper bounds in Theorems 1.1, 1.2, and 1.3. The section is organized as follows. In the first subsection we show that for every almost-regular graph that is far from being k-colorable, a random induced subgraph of size $O\left(\frac{n}{d}\right)$ is not k-colorable with high probability. In the second subsection we give efficient procedures for sampling edges almost uniformly in general graphs, and in the third subsection we give a framework for converting the bounds from general graphs to almost-regular graphs. In the last subsection we show how to implement the reduction framework using various query types.

3.1. Almost Regular Graphs. In this subsection we prove the following theorem.

THEOREM 3.1. *Let G be an almost-regular graph on n vertices with average degree d. If G is ϵ-far from being k-colorable for a constant ϵ and $k \geq 3$, then a random induced subgraph of size $\Theta\left(\frac{n}{d}\right)$ is not k-colorable with high probability.*

The proof of Theorem 3.1 extends arguments presented in [4]. Details follow.

Let $G = (V, E)$ be an almost-regular graph over n vertices with average degree d (so that the degree of every vertex in the graph is at most $10d$). Given a subset $S \subset V$ and a k-coloring $\varphi : S \to [k]$ of S, for every $v \in V \setminus S$ let

$$L_{(S,\varphi)}(v) \stackrel{\text{def}}{=} [k] \setminus \{1 \leq i \leq k : \ \exists u \in S \cap \Gamma(v), \varphi(u) = i\} \ .$$

Note that if $S = \emptyset$, then $L_{(S,\varphi)}(v) = [k]$ for every $v \in V$. Observe that if a legal k-coloring $c : V(G) \to [k]$ of G coincides with φ on S, then for every $v \in V \setminus S$ the color of v in c belongs to $L_{(S,\varphi)}(v)$. Hence, $L_{(S,\varphi)}(v)$ is called the *list of feasible colors* for v with respect to (S, φ). A vertex $v \in V \setminus S$ is called *colorless* with respect to (S, φ), if $L_{(S,\varphi)}(v) = \emptyset$. We denote by $U_{(S,\varphi)}$ the set of all colorless vertices with respect to (S, φ). For every vertex $v \in V \setminus (S \cup U_{(S,\varphi)})$ define

$$\delta_{(S,\varphi)}(v) \stackrel{\text{def}}{=} \min_{i \in L_{(S,\varphi)}(v)} |\{u \in \Gamma(v) \setminus (S \cup U_{(S,\varphi)}) : \ i \in L_{(S,\varphi)}(u)\}| \ .$$

Thus, coloring v by one of the colors from $L_{(S,\varphi)}(v)$ and then adding it to S results in deleting this color from the lists of feasible colors of at least $\delta_{(S,\varphi)}(v)$ neighbors of v outside S.

LEMMA 3.2. *For every set $S \subset V$ and every k-coloring φ of S, the number of edges that must be removed from G so that it become k-colorable is at most $10d|S \cup U_{(S,\varphi)}| + \sum_{v \in V \setminus (S \cup U_{(S,\varphi)})} \delta_{(S,\varphi)}(v)$.*

Proof. For every $v \in S$, color v by $\varphi(v)$. For every $v \in U_{(S,\varphi)}$, color v by an arbitrary color from $[k]$. For every $v \in V \setminus (S \cup U_{(S,\varphi)})$, color v with a color $i \in L_{(S,\varphi)}(v)$ for which $\delta_{(S,\varphi)}(v) = |\{u \in \Gamma(v) \setminus (S \cup U_{(S,\varphi)}) : \ i \in L_{(S,\varphi)}(u)\}|$.

We next upper bound the number of monochromatic edges according to this coloring. The number of monochromatic edges incident with $S \cup U_{(S,\varphi)}$ is at most $10d|S \cup U_{(S,\varphi)}|$ (recall that the graph is almost-regular). Every vertex $v \in V \setminus (S \cup U_{(S,\varphi)})$ has exactly $\delta_{(S,\varphi)}(v)$ neighbors $u \in V \setminus (S \cup U_{(S,\varphi)})$, whose color list $L_{(S,\varphi)}(v)$ contains the color chosen for v. Therefore, v has at most $\delta_{(S,\varphi)}(v)$ neighbors in $V \setminus (S \cup U_{(S,\varphi)})$ colored with the same color. Hence, the total number of monochromatic edges is as claimed. $\square$

As an immediate corollary of Lemma 3.2 we get:

Corollary 3.3. *If G is an almost-regular graph that is ϵ-far from being k-colorable, then for any pair (S, φ), where $S \subset V(G)$, $\varphi : S \to [k]$:*

$$\sum_{v \in V \setminus (S \cup U_{(S,\varphi)})} \delta_{(S,\varphi)}(v) > \epsilon dn - 10d(|S| + |U_{(S,\varphi)}|) \ .$$

Given a pair (S, φ), a vertex $v \in V \setminus (S \cup U_{(S,\varphi)})$ is called *restricting* if $\delta_{(S,\varphi)}(v) \geq \epsilon d/2$. We denote by $W_{(S,\varphi)}$ the set of all restricting vertices.

Lemma 3.4. *If G is an almost-regular graph that is ϵ-far from being k-colorable, then for every pair (S, φ), where $S \subset V(G)$, $\varphi : S \to [k]$, we have:*

$$|U_{(S,\varphi)} \cup W_{(S,\varphi)}| > \epsilon n/20 - |S|.$$

Proof. By Corollary 3.3,

$$\begin{aligned}
\epsilon dn - 10d(|S| + |U_{(S,\varphi)}|) &< \sum_{v \in V \setminus (S \cup U_{(S,\varphi)})} \delta_{(S,\varphi)}(v) \\
&\leq 10d|W_{(S,\varphi)}| + \sum_{v \in V \setminus (S \cup U_{(S,\varphi)} \cup W_{(S,\varphi)})} \delta_{(S,\varphi)}(v) \\
&< 10d|W_{(S,\varphi)}| + \frac{n \cdot \epsilon d}{2} \ .
\end{aligned}$$

This implies that $|S| + |U_{(S,\varphi)}| + |W_{(S,\varphi)}| \geq \epsilon n/20$. As $U_{(S,\varphi)}$ and $W_{(S,\varphi)}$ are disjoint, the lemma follows. $\square$

Constructing an auxiliary k-ary tree. Consider an almost-regular graph G (with n vertices and average degree d), that is ϵ-far from being k-colorable. While choosing random vertices $r_1, \cdots, r_s$ we construct an auxiliary k-ary tree T. To distinguish between the vertices of G and those of T we call the latter *nodes*. In the course of the construction, each internal node of T is labeled by a vertex of G (among the random vertices $r_1, \cdots, r_s$). Each leaf is either labeled by a special symbol #, in which case it is called a *terminal node*, or it is unlabeled (and may later become an internal node). Different internal nodes in the tree may be labeled by the same vertex in G. However, on each path from the root to a leaf the sequences of labels are distinct. The edges of T are labeled by integers from $[k]$ (colors). We start the construction of T from an unlabeled single node, the root of T (which is initially also a leaf).

Let y be a node of T, and consider the path from the root of T to y. Let the nodes on the path be $z_0, z_1, \ldots, z_\ell = y$ (where z_0 is the root). For $0 \leq j \leq \ell - 1$, let the label of z_j be v_j, and let the label of the edge between z_j and z_{j+1} be i_j. We denote the set $\{v_0, \ldots, v_{\ell-1}\}$ by S_y, and the coloring of S_y induced by the labels of the edges, by φ_y. That is, $\varphi_y(v_j) = i_j$. The labeling of the nodes and edges of T will have the following property: if a node y in T is labeled by a vertex v in G and v has a neighbor in S_y whose color according to φ_y is i, then the child of y along the edge labeled by i, is labeled by # (i.e., it is a terminal node). This label indicates the fact that given (S_y, φ_y), color i is infeasible for v.

We think of selecting the vertices $r_1, \cdots, r_s$ in s *rounds*. Suppose that $j-1$ vertices $r_1, \ldots, r_{j-1}$ have already been selected, and we select (uniformly at random) a vertex r_j. For each node y that is currently a leaf of T, if y is labeled by #, we do nothing for this leaf. (This is the reason such a node y is called a terminal node; nothing will ever grow out of this node.) Assume now that y is unlabeled. Define the pair (S_y, φ_y) as described above. For the pair (S_y, φ_y), we let U_y be a shorthand for $U_{(S_y,\varphi_y)}$ (the colorless vertices with respect to (S_y, φ_y)) and we let W_y be a shorthand for $W_{(S_y,\varphi_y)}$ (the restricting vertices with respect to (S_y, φ_y)).

Round j is called *successful* for the node y, if the random vertex r_j satisfies $r_j \in U_y \cup W_y$. If round j is indeed successful for y, then we do the following. We label y by r_j, create k children of y and label the corresponding edges by $1, \cdots, k$. If color i is infeasible for r_j given (S_y, φ_y) (that is, $i \notin L_{(S_y,\varphi_y)}$), then we label the child of y along the edge with label i by #. Otherwise we leave this child (which is currently a leaf) unlabeled. Note that if $r_j \in U_y$, then none of the colors from $[k]$ is feasible for r_j, and thus all the children of y will be labeled by #. If round j is not successful for y, then y remains an unlabeled leaf. This completes the description of the process of constructing T. Next, we state some properties of T.

The first claim follows directly from the definition of the labeling procedure.

CLAIM 3.5. *If a leaf z^* of T is labeled by #, then φ_{z^*} is not a proper k-coloring of S_{z^*}.*

LEMMA 3.6. *If after round j all leaves of the tree T are terminal nodes, then the subgraph $G[\{r_1, \cdots, r_j\}]$ is not k-colorable.*

PROOF. Note that by the construction of the tree, each internal node of T is labeled by a vertex in $\{r_1, \cdots, r_j\}$ (and each leaf is either labeled by # or is unlabeled). Let $c : \{r_1, \cdots, r_j\} \to [k]$ be a k-coloring of $\{r_1, \cdots, r_j\}$. In order to show that c induces some monochromatic edges in the induced subgraph of G on $\{r_1, \cdots, r_j\}$, we start with the root z_0 of T and traverse T guided by c as follows: while at node y of T, labeled by $v(y) \in \{r_1, \cdots, r_j\}$, we move from y to its child along the edge of T labeled by $c(v(y))$. Once we reach a terminal node z^* of T, we have that $S_{z^*} \subseteq \{r_1, \cdots, r_j\}$ and φ_{z^*} coincides with c on S_{z^*}. As z^* is a terminal node, it follows from Claim 3.5 that c is not a proper k-coloring of S_{z^*}. $\square$

CLAIM 3.7. *The depth of T is bounded from above by $\frac{2kn}{\epsilon d} + 1$.*

PROOF. Let z^* be a leaf of T. Recall that if the label of a node y of T belongs to U_y, then all children of y in T are labeled by # and are terminal nodes. Therefore, for each node y on the path from the root of T to z^*, but possibly the node immediately preceding z^*, the label of y belongs to W_y. Since each vertex in W_y is restricting with respect to (S_y, φ_y), coloring y in any feasible color (in $L_{(S_y,\varphi_y)}$) decreases the total size of the lists of feasible colors for all vertices of G by at least $\epsilon d/2$. Therefore, each time when on the path from the root of T to z^* we leave a node y, whose label belongs to W_y, the total length of the list of feasible colors decreases by at least $\epsilon d/2$. As initially all k colors are feasible for all vertices, we start with lists of feasible colors of total size nk. Thus, we cannot make more than $nk/(\epsilon d/2) + 1 = \frac{2nk}{\epsilon d} + 1$ steps down from the root of T to z^*. This implies that the depth of T is at most $\frac{2nk}{\epsilon d} + 1$. $\square$

Lemma 3.8. *For a graph G with average degree $d \geq 120k/\epsilon^2$, if G is ϵ-far from being k-colorable, then with high probability, after $s = \Theta\left(\frac{k \log kn}{\epsilon^2 d}\right)$ rounds, all leaves of T are terminal nodes.*

Proof. Let T' denote the complete k-ary tree of depth $\frac{2kn}{\epsilon d} + 1$. Since every internal node of T has k children and (by Claim 3.7) T has depth at most $\frac{2kn}{\epsilon d} + 1$, it is a subgraph of T'. In particular, the number of leaves in T', is upper bounded by $k^{\frac{2kn}{\epsilon d}+1}$. We shall prove that with high probability over the choice of $R = \{r_1, \ldots, r_s\}$, for every leaf z^* of T', either z^* or one of its ancestors in T' becomes a terminal node during the construction of T. This is equivalent to the statement in the lemma.

Consider any fixed choice of a leaf z^* in T', and let $z_0, \ldots, z_{\ell-1}, z_\ell = z^*$ be the path from the root of T' to z^*. For each $1 \leq j \leq s$, when r_j is selected, some prefix $z_0, \ldots, z_{t-1}$ of this path has been labeled (in T) by a subset of vertices $S_{z_t} \subseteq \{r_1, \ldots, r_{j-1}\}$ (if the prefix is empty, then $z_t = z_0$ and $S_{z_t} = \emptyset$), and z_t is currently a leaf in T. Suppose that z_t is not labeled by #. Let U_{z_t} be a shorthand for $U_{(S_{z_t}, \varphi_{z_t})}$ and let W_{z_t} be a shorthand for $W_{(S_{z_t}, \varphi_{z_t})}$. By Lemma 3.4, $|U_{z_t} \cup W_{z_t}| > \epsilon n/20 - |S_{z_t}| = \epsilon n/20 - t$, and by Claim 3.7, $t \leq \frac{2kn}{\epsilon d} + 1$. By the premise of the lemma, $d > 120k/\epsilon^2$, and so $|U_{z_t} \cup W_{z_t}| > \epsilon n/40$. Therefore, the probability that $r_j \in U_{z_t} \cup W_{z_t}$, so that the round is successful for z_t, is at least $\epsilon/40$.

For each $1 \leq j \leq s$ we define a Bernoulli random variable $X_j(z^*)$. The value of $X_j(z^*)$ is 1 if and only if one of the following holds: (1) When r_j is selected, some node z_t on the path from the root of T' to z^* is already labeled #; or (2) When selecting r_j, the deepest unlabeled node on the path (currently a leaf) is z_t and $r_j \in U_{z_t} \cup W_{z_t}$. By the discussion above, $\Pr[X_j(z^*) = 1] > \epsilon/40$ (where if the first condition holds, then we actually have that $\Pr[X_j(z^*) = 1] = 1$). We are interested in upper bounding the probability that $\sum_{j=1}^{s} X_j(z^*) \leq \frac{2nk}{\epsilon d}$ when $s = \frac{ck \ln kn}{\epsilon^2 d}$ for a sufficiently large constant c. While these random variables are not independent, the probability that $X_j(s^*) = 1$ is at least $\epsilon/40$ conditioned on *every* setting of $X_1(s^*), \ldots, X_{j-1}(s^*)$. Therefore, the probability of this event is upper bounded by the probability that for s *independent* Bernoulli random variables $Y_1, \ldots, Y_s$ such that $\Pr[Y_j = 1] = \epsilon/40$, we get that $\sum_{j=1}^{s} Y_j \leq \frac{2nk}{\epsilon d}$. For $s > \frac{160k \ln kn}{\epsilon^2 d}$, this is less than half the expected value of $\sum_{j=1}^{s} Y_j$. Therefore, by a multiplicative Chernoff bound,

$$\Pr\left[\sum_{j=1}^{s} Y_j \leq \frac{2nk}{\epsilon d}\right] < \exp\left(-\frac{1}{8}\cdot\frac{\epsilon}{40}\cdot s\right)$$
$$= \exp\left(-\frac{c}{320}\cdot\frac{k\ln kn}{\epsilon d}\right)$$
$$= k^{-\frac{c}{320}\cdot\frac{kn}{\epsilon d}}.$$

By setting c to be a sufficiently large constant, if we now take a union bound over all $k^{\frac{2kn}{\epsilon d}+1}$ leaves of T', the probability that for some leaf it, neither it, or any of its ancestors is a terminal node, is $o(1)$, as claimed. $\square$

Proof of Theorem 3.1. The theorem follows by combining Lemma 3.6 with Lemma 3.8. $\square$

3.2. Sampling edges revisited. The problem of sampling edges plays a significant role in the context of property testing in general graphs, and in particular in this work. It is very easy to sample an edge using $O\left(\frac{n}{d}\right)$ pair queries, by randomly selecting pairs of vertices until we get an edge. However, we need a more efficient way, and thus we relax our requirements. Given a parameter $0 < \delta < 1/4$, we would like to select each edge, apart from a set of edges of size at most δdn, with probability between $\frac{c_1}{dn}$ and $\frac{c_2}{dn}$, for some fixed constants c_1 and c_2. The procedure may fail to output an edge, and we bound the failure probability by $O\left(\frac{1}{n^2}\right)$. Since one can trivially sample edges uniformly by checking all the degrees of the vertices in advance using n degree queries, we may assume that such an edge-sampling procedure is called less than n times by any testing algorithm. Therefore, the probability that during the algorithm's execution the procedure will fail to return an edge in any one of its calls is $o(1)$. We say that such a procedure samples edges *δ-uniformly*. If δ is a small constant, then we shall say that the procedure samples edges *almost-uniformly*. For $t > 1$, we shall say that a (multi-)set of edges $e_1, \ldots, e_t$ is selected δ-uniformly if for every $1 \leq j \leq t$, the edge e_j is selected δ-uniformly (conditioned on any choice of $e_1, \ldots, e_{j-1}$).

In [18] it was shown how to sample an edge δ-uniformly using $O\left(\sqrt{\frac{n}{\delta}}\right)$ neighbor and degree queries. Here we present two algorithms that use neighbor and degree queries and sample edges δ-uniformly. The first one uses $\tilde{O}\left(\sqrt{\frac{n}{\delta d}}\right)$ queries (thus improving on the result in [18] when d is large), while the second one samples t edges using $\tilde{O}\left(\frac{n}{\delta d}+t\right)$ queries. We note that we shall not actually use the first procedure (for sampling single edges), but we include it since it may be useful in other contexts.

LEMMA 3.9. *There exists a procedure that samples an edge δ-uniformly and such that the expected number of (random) neighbor and degree queries that it performs is $O\left(\sqrt{\frac{n}{\delta d}}\log n\right)$.*

PROOF. We say that a vertex has *high degree* if its degree is at least $c\sqrt{nd}$, where $c = \sqrt{\frac{1}{2\delta}}$. The number of high degree vertices is at most $\frac{nd}{c\sqrt{nd}} = \frac{\sqrt{nd}}{c}$. We say that an edge is *bad* if both of its end-points have high degree. It follows that the number of bad edges is at most $\frac{nd}{2c^2} = \delta nd$.

We repeat the following step until an edge is selected. Choose a vertex v randomly, and check its degree, denote it by $d(v)$. With probability $\frac{\min\{d(v), c\sqrt{dn}\}}{c\sqrt{dn}}$ select a random edge incident to v, output it, and terminate. (With probability $1 - \frac{\min\{d(v), c\sqrt{dn}\}}{c\sqrt{dn}}$ continue.) For every edge $e = (u, v)$, the probability p_e that e is picked at a certain step is

$$p_e = \frac{1}{cn\sqrt{dn}}\left(\frac{\min\{d(v), c\sqrt{dn}\}}{d(v)} + \frac{\min\{d(u), c\sqrt{dn}\}}{d(u)}\right) .$$

Therefore, every edge e satisfies $p_e \leq \frac{2}{cn\sqrt{dn}}$, and every non-bad edge satisfies $p_e \geq \frac{1}{cn\sqrt{dn}}$. Since the steps are independent, conditioned on a step outputting an edge, for all but at most δdn edges, the probability that any particular edge is output is $\Theta\left(\frac{1}{dn}\right)$. as required.

It remains to show that the procedure terminates (and outputs an edge) after an expected number of $O\left(\sqrt{\frac{n}{\delta d}}\right)$ steps. To this end we have

$$\sum_{v\,:\,d(v)\leq c\sqrt{dn}} d(v) \geq nd\left(\frac{1}{2} - \delta\right) .$$

As $\delta < 1/4$, the probability that the procedure terminates in a certain step is at least:

$$\frac{1}{n} \cdot \frac{\sum_{v\,:\,d(v)\leq c\sqrt{dn}} d(v)}{c\sqrt{dn}} \geq \frac{1}{n} \cdot \frac{nd}{4c\sqrt{dn}} = \Omega\left(\sqrt{\frac{\delta d}{n}}\right) .$$

Thus the probability that the number of steps exceeds $\alpha\sqrt{\frac{n}{\delta d}}$ is $O\left(\frac{1}{n^2}\right)$ for $\alpha = O(\log n)$, and the claim follows. $\square$

LEMMA 3.10. *There exists a procedure that given as input an integer t and $\delta > 0$, performs $O\left(\frac{n\log n}{\delta d} + t\right)$ degree queries and (random) neighbor queries and outputs t edges, where with high probability the t edges are selected δ-uniformly.*

PROOF. The procedure works in two phases. In the first phase it selects, uniformly, independently and at random (with replacement), $s = \Theta\left(\frac{n\log n}{\delta d}\right)$ vertices. Let S denote the multi-set of vertices selected. The procedure queries the degree of each vertex u in S, and computes $d(S) = \sum_{u\in S} d(u)$ (since S is a multiset, if a vertex u appears r times in the multi-set, then it contributes $r \cdot d(u)$ to $d(S)$). In the second phase the procedure repeats the following t times:

1. Select a vertex u in S with probability $d(u)/d(S)$.
2. Select a random neighbor v of u.
3. Select a random edge e incident to v, and output e.

Thus, the total number of queries performed is $O\left(\frac{n\log n}{\delta d} + t\right)$ as required. We next prove that with high probability over the choice of S, each edge selected in the second phase is δ-uniformly distributed.

For any vertex v, let $d_S(v)$ denote the number of neighbors that v has in S (once again, including repetitions), so that the expected value of $d_S(v)$ is $\frac{s}{n} \cdot d(v)$. By a multiplicative Chernoff bound, for each fixed vertex v such that $d(v) \geq \delta d$, we have that (for an appropriate constant in the $\Theta(\cdot)$ notation for s) with probability at least $1 - n^{-2}$,

$$\frac{1}{2} \cdot \frac{s}{n} \cdot d(v) \;\leq\; d_S(v) \;\leq\; 2 \cdot \frac{s}{n} \cdot d(v)\,. \tag{3.11}$$

Similarly, for each fixed vertex v such that $d(v) < \delta d$, with probability at least $1 - n^{-2}$,

$$d_S(v) \;\leq\; 2 \cdot \frac{s}{n} \cdot 2\delta d\,. \tag{3.12}$$

By taking a union bound over all vertices, we get that with probability at least $1 - 1/n$ over the choice of S, Equation (3.11) holds for every v such that $d(v) \geq \delta d$, and Equation (3.12) holds for every v such that $d(v) < \delta d$. We shall say in such a case that S is *degree-representative*.

Observe that the total number of edges incident to vertices with degree less than δd is at most δdn, and indeed, such edges might be selected with very small probability. We next show that if S is degree-representative, then for every edge $e = (v, w)$ such that $d(v) \geq \delta d$ and $d(w) \geq \delta d$, the probability that e is selected in any particular step of the second phase is $\Theta\left(\frac{1}{dn}\right)$. Let us denote this event by $E_{(v,w)}$, and for any vertex y, let E_y denote the event that y is selected in the second substep of an edge selection step (i.e., when selecting a random neighbor of a selected vertex $u \in S$). For any vertex y,

$$\Pr[E_y] \;=\; \sum_{u \in \Gamma(y) \cap S} \frac{d(u)}{d(S)} \cdot \frac{1}{d(u)} \;=\; \frac{d_S(y)}{d(S)} \;.$$

Using the premise that S is degree-representative, and the fact that $d(S) = \sum_{u \in S} d(u) = \sum_y d_S(y)$,

$$\begin{aligned}
\Pr[E_{(v,w)}] &= \frac{1}{d(v)} \cdot \Pr[E_v] + \frac{1}{d(w)} \cdot \Pr[E_w] \\
&= \frac{1}{d(v)} \cdot \frac{d_S(v)}{d(S)} + \frac{1}{d(w)} \cdot \frac{d_S(w)}{d(S)} \\
&\leq \frac{1}{d(v)} \cdot \frac{2(s/n)d(v)}{d(S)} + \frac{1}{d(w)} \cdot \frac{2(s/n)d(w)}{d(S)} \\
&= \frac{4(s/n)}{\sum_{y:d(y)\geq \delta d} d_S(y) + \sum_{y:d(y)<\delta d} d_S(y)} \\
&\leq \frac{4(s/n)}{\sum_{y:d(y)\geq \delta d} (1/2)(s/n)d(y)} \\
&\leq \Theta\left(\frac{1}{dn}\right) \;.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\Pr[E_{(v,w)}] &\geq \frac{1}{d(v)} \cdot \frac{(1/2)(s/n)d(v)}{d(S)} + \frac{1}{d(w)} \cdot \frac{(1/2)(s/n)d(w)}{d(S)} \\
&= \frac{s/n}{\sum_{y:d(y)\geq \delta d} d_S(y) + \sum_{y:d(y)<\delta d} d_S(y)} \\
&\geq \frac{s/n}{\sum_{y:d(y)\geq \delta d} 2(s/n)d(y) + \sum_{y:d(y)<\delta d} 4(s/n)\delta d} \\
&= \Theta\left(\frac{1}{dn}\right) \;,
\end{aligned}$$

and the lemma follows. $\square$

3.3. A reduction from the general case. We now describe a reduction from general graphs to almost-regular graphs. It was proved in [18] that for every graph G with average degree d it is possible to construct (using a probabilistic procedure) a graph G' that is almost-d-regular and has several additional properties that are useful in the context of testing bipartiteness. Here we extend the result to the problem of testing k-colorability. In what follows we refer to almost-uniform sampling of vertices, where this notion of sampling of vertices is analogous to the one defined for edges in Subsection 3.2.

Lemma 3.13. *For every graph G on n vertices and average degree $d = \Omega(\log n)$, and for every $k \geq 3$, we can construct randomly a graph G' that has the following properties with high probability:*

(i) G' has at most $2n$ vertices, the same number of edges as G, and maximal degree $d' < 2d$.

(ii) If G is k-colorable then G' is also k-colorable.

(iii) If G is ϵ-far from being k-colorable then G' is ϵ'-far being from being k-colorable for $\epsilon' = \Theta(\epsilon)$.

(iv) It is possible to sample an induced subgraph of size $\tilde{O}(\frac{n}{d})$ in G' where the vertices of the subgraph are selected δ-uniformly for $\delta = \Omega(\epsilon)$, using either $\tilde{O}(\frac{n}{\epsilon d})$ group queries or $\tilde{O}((\frac{n}{d})^2)$ pair queries on G.

Proof: Every vertex v of G is transformed into $\lceil \deg(v)/d \rceil$ vertices in G'. Denote by $X(v)$ the vertices in G' related to a vertex $v \in V(G)$. The vertices in $X(v)$ are denoted by $X_i(v)$, $1 \leq i \leq \lceil \deg(v)/d \rceil$. Thus, $n' = |V(G')| \leq \sum_{v \in G} \lceil \deg(v)/d \rceil \leq 2n$. The edges of G' are determined as follows: an edge $(u, v) \in E(G)$ chooses independently uniformly at random a vertex from $X(v)$ and a vertex from $X(u)$. In G' there will be an edge between these two randomly chosen vertices. Clearly, $|E(G')| = |E(G)| = (nd)/2$. The required properties of G' follow from the claims below.

Claim 3.14. *For $d = \Omega(\log n)$, the maximum degree d' of G' constructed above is at most $2d$ with probability $1 - o(1)$.*

The proof of Claim 3.14 can be found as Lemma 7 in [18].

CLAIM 3.15. *For a graph G with $d > 17k^4/\epsilon$ the following holds: If G is ϵ-far from being k-colorable, then with probability $1 - o(1)$, G' is ϵ'-far from being k-colorable for $\epsilon' = \frac{\epsilon}{16k^3}$.*

PROOF. Consider a fixed k-partition $P' = (V'_1, \cdots, V'_k)$ of the vertices in G'. The partition P' induces a partition of $X(v)$ for every v. Let us denote by $X^\alpha(v)$ the largest subset of $X(v)$ induced by P', breaking ties arbitrarily. For a graph T and $U \subset V(T)$, denote by $E(U)$ the set of edges in the subgraph of T induced by U. Consider a partition $P = (V_1, \ldots, V_k)$ of the vertices of G induced by P' in the following way. For $v \in V(G)$, if $X^\alpha(v) \subset V'_i$ for $1 \le i \le k$, then $v \in V_i$. Since G is ϵ-far from being k-colorable, for at least one of the subsets $V_1, \cdots, V_k$, the induced subgraph contains at least $\frac{1}{k}\epsilon dn$ edges. Without loss of generality assume that $|E(V_1)| \ge \frac{1}{k}\epsilon dn$. Let H' be a subgraph of G', defined as follows. The vertices of H' are $\bigcup_{v \in V_1} X^\alpha(v)$, and the edges of H' are the edges of G', induced by $V(H')$. Thus, $V(H') \subset V'_1$ and $E(H') \subset E(V'_1)$.

We next show the following for $c > 1$ (using our lower-bound assumption on d):

$$\Pr[|E(H')| \le (1/4k^3)\epsilon dn] < k^{-c \cdot n'} \tag{3.16}$$

Once we establish Inequality (3.16), by taking the union bound over all possible partitions P' of the vertices of G' we get that for *every* partition $P' = (V'_1, \cdots, V'_k)$ of the vertices of G', the number of violating edges in G' (with respect to P'), is at least $(1/4k^3)\epsilon dn$ with probability $1 - o(1)$. Recall that $n' \le 2n$ and $d' \le 2d$ with probability $1 - o(1)$. Thus, for every partition of the vertices of G', the number of violating edges is at least $(\epsilon/16k^3) \cdot n' \cdot d'$ with probability $1 - o(1)$, as required.

Proof of Inequality (3.16): Consider an edge $e = (u, v) \in E(G)$, and let $\varphi(e) = (X_i(v), X_j(u))$ be its corresponding edge in $E(G')$. For $e \in E(V_1)$, $\Pr[\varphi(e) \in E(H')] \ge (1/k) \cdot (1/k) = 1/k^2$. Thus,

$$\mathrm{Exp}[|E(H')|] \ge \frac{1}{k^2}|E(V_1)|.$$

As $|E(H')|$ is a sum of $|E(V_1)|$ independent Bernoulli random variables, each with expectation at least $1/k^2$, it follows from Chernoff bounds that

$$\Pr[|E(H')| < (1/4k^3)\epsilon dn] < e^{-\epsilon dn/16k^3}$$

Thus, for $d \ge 16k^4/\epsilon$ we have

$$\Pr[|E(H')| \le (1/4k^3)\epsilon dn] < k^{-cn}$$

for $c > 1$, and the claim is proved. □

Claim 3.17. *With high probability, an induced subgraph of G' on t almost-uniformly selected vertices contains $\tilde{O}(\frac{t^2 d}{n})$ edges.*

Proof. Let $S \subseteq V$ be a (multi-)set of size t selected δ-uniformly. For every edge $e = (u, v)$ of G', the probability that both u and v fall into S is $O\left(\frac{t^2}{n^2}\right)$. By the linearity of expectation, the expected total number of edges in such an induced subgraph is $O\left(\frac{ndt^2}{n^2}\right) = O\left(\frac{t^2 d}{n}\right)$. The desired bound follows by the Markov inequality. □

Claim 3.18. *It is possible to sample an induced subgraph of size $\tilde{O}(\frac{n}{d})$ in G' whose vertices are selected δ-uniformly for $\delta = \Omega(\epsilon)$, using either $\tilde{O}(\frac{n}{\epsilon d})$ group queries or $\tilde{O}((\frac{n}{d})^2)$ pair queries on G.*

Proof. In order to query an induced subgraph as specified in the claim we first need to select $\tilde{O}(\frac{n}{d})$ random vertices δ-uniformly and then to find all the edges between them. By Claim 3.17, such an induced subgraph contains $\tilde{O}(\frac{n}{d})$ edges with high probability. As every vertex in G has a number of copies that is proportional to its degree, selecting a vertex in G' δ-uniformly reduces to selecting an edge in G δ-uniformly. We can sample an edge uniformly using $O\left(\frac{n}{d}\right)$ pair queries, and by Lemma 3.10 we can sample δ-uniformly $\tilde{O}(\frac{n}{d})$ random edges using $\tilde{O}(\frac{n}{\delta d})$ random neighbor queries and degree queries. By Claim 2.1, random neighbor queries can be emulated by group queries. Therefore, for the first step it suffices to perform either $\tilde{O}(\frac{n}{\epsilon d})$ group queries and degree queries or $\tilde{O}((\frac{n}{d})^2)$ pair queries.

For the second step, in the pair query model we just query all the relevant pairs in the original graph. For every pair of connected vertices in G we choose randomly which of their copies are connected in G', and according to the answers recover the induced subgraph in G'. Similarly, in the group query model we first use degree queries on all the vertices. By Claim 2.2, it is possible to recover the induced subgraph using $\tilde{O}(\frac{n}{d})$ group queries. The only modification is that we need to decide, before we start the step, for every pair of relevant vertices in G, which of their copies are connected (in the case that they are actually connected in G). Then, according to these decisions, we emulate the group queries on G' by group queries on G. This completes the proof. □

(Claim 3.18 and Lemma 3.13)

3.4. Establishing the upper bounds. In this subsection we show how to use the reduction (Lemma 3.13) to establish bounds for the pair query model and the group query model. In the neighbor query model we give an upper bound using a different (and simpler) approach.

Proof of Item 1 in Theorem 1.2. Our algorithm has two steps : In the first step, we perform a degree query on every vertex of the graph. Using this information, we can now choose a random edge with a single query (just choose a vertex with probability proportional to its degree, and choose a random edge incident to this vertex). In the second step sample $\frac{n \ln k}{2\epsilon} + \frac{1}{\epsilon}$ edges uniformly and independently. If the graph is k-colorable then trivially every subgraph will be k-colorable. On the other hand, if the graph is ϵ-far from being k-colorable, then every fixed k-partition has at least ϵdn violating edges. As the total number of edges is $\frac{nd}{2}$, the probability that none of the violating edges was selected is at most

$$(1-2\epsilon)^{\frac{n \ln k}{2\epsilon}+\frac{1}{\epsilon}} < e^{-2} \cdot e^{-n \ln k} < \frac{1}{3} k^{-n} .$$

Using the union bound, with probability at least $2/3$ every partition will have a violating edge in the sample, and thus the sample will not be k-colorable. □

Proof of Item 1 in Theorem 1.1 and Item 1 in Theorem 1.3. Consider first the case that $d = o(\log n)$. For this case, in the pair query model just query all the pairs of vertices in the graph. In the group query model use the tester of the neighbor query model, emulating neighbor queries using group queries (Claim 2.1). Otherwise ($d = \Omega(\log n)$), given a graph G, let G' be the graph obtained by Lemma 3.13. By Lemma 3.13, It is possible to sample an induced subgraph of size $\tilde{O}(\frac{n}{d})$ in G' where the vertices of the subgraph are selected δ-uniformly, using either $\tilde{O}(\frac{n}{\delta d})$ group queries or $\tilde{O}((\frac{n}{d})^2)$ pair queries on G. By Theorem 3.1, in order to test k-colorability it suffices to uniformly sample an induced subgraph of size $\tilde{O}(\frac{n}{d})$ in G'. While Theorem 3.1 is stated for uniform sampling, it also holds for δ-uniform sampling when $\delta = \epsilon/c$ for a sufficiently large constant c. This can be verified by observing that the only place where the distribution on selected vertices plays a role in the proof of Theorem 3.1 is in the proof of Lemma 3.8. This proof relies on lower bounding the probability of hitting various sets vertices having size $\Omega(\epsilon n)$, and hence holds also when sampling δ-uniformly for $\delta = \epsilon/c$ and a sufficiently large constant c.

4. Lower Bounds

In this section we prove the lower bounds in Theorems 1.1, 1.2, and 1.3. In the first subsection we give several building blocks used in the lower bounds. In particular we describe a probabilistic construction of a graph that is far from being k-colorable yet every induced subgraph of linear size is 3-colorable. In the second subsection we establish the lower bounds in the various models.

4.1. The Building Blocks.

4.1.1. A Probabilistic Construction. We first would like to generate a sparse graph that is far from being k-colorable, yet every induced subgraph of linear size is 3-colorable. We show that a graph from the distribution $G(n,p)$ satisfies these conditions with high probability for an appropriate choice of p. In the proof we follow ideas of Erdős (see [11]).

Lemma 4.1. *For every constant value of $k \geq 3$ there exist constants $\alpha(k), c(k) > 0$ such that the following properties hold with high probability for a random graph H selected according to the distribution $G(n, p = \frac{c(k)}{n})$.*

- *H has $\Theta(n)$ edges.*
- *H is $\Theta(1)$-far from being k-colorable.*
- *Every induced subgraph of H on $\alpha(k)n$ vertices is 3-colorable.*

Proof. Consider a graph H selected according to the distribution $G(n, p = 8k^3/n)$. We show that with high probability, H satisfies all the items in Lemma 4.1.

The number of edges in H is binomially distributed with parameters $\binom{n}{2}$ and p. Using Chernoff type bounds, for any $\delta > 0$ and large enough n, with probability $1 - o(1)$, the graph contains $(4 \pm 2\delta)k^3 n$ edges, and this proves the first item.

Next, we show that H is far from being k-colorable with probability $1-o(1)$. Consider a k-partition $P = (V_1, \ldots, V_k)$ of the n vertices. The number of pairs of vertices with both end-points in the same subset V_i is $\sum_{i=1}^{k} \binom{|V_i|}{2}$. By the convexity of the function $\binom{x}{2}$, this number is at least $k\binom{\frac{n}{k}}{2}$. Therefore, the expected number of violating edges with respect to the partition P is at least $k\binom{\frac{n}{k}}{2} \cdot 8k^3/n = 4k^2 n(1-o(1))$. By the Chernoff bound, the probability that the number of violating edges is less than $2k^2 n$ (roughly half the expected value) is upper bounded by $\exp(-\Omega((k^3/n) \cdot (n^2/k))) = \exp(-\Omega(k^2 n)) \ll k^{-n}$. By taking the union bound over all possible partitions (whose number is upper bounded by k^n), we get that with probability $1 - o(1)$, for *every* partition of the n vertices, the number of violating edges in H is at least $2k^2 n$ and the graph is $\frac{1}{(2+\delta)k}$-far from being k-colorable.

To prove the third item, we show that with probability $1 - o(1)$, for $\alpha = \alpha(k) < \frac{1}{55^2 k^9}$, every subset of vertices of H of size αn is 3-colorable. Recall that a k-critical graph is a k-colorable graph with the property that every proper subgraph is $(k-1)$-colorable. Note that for a subset of vertices $S \subset V(H)$, if the subgraph, $H[S]$, induced by S is not 3-colorable, then it must contain a 4-critical subgraph, and every such graph has minimum degree at least 3. This implies that there is $S' \subseteq S$ such that $|E(H[S'])| \geq 3|S'|/2$. The probability that there exists a set $S \subset V(H)$ of size $s \leq \alpha n - 1$ such that $|E(H[S])| \geq 3s/2$ is upper bounded by

$$\sum_{s=4}^{\alpha n-1} \binom{n}{s} \binom{\binom{s}{2}}{3s/2} (8k^3/n)^{3s/2} . \tag{4.2}$$

Consider a single term in the sum:

$$\begin{aligned} \binom{n}{s} \binom{\binom{s}{2}}{3s/2} (8k^3/n)^{3s/2} &\leq \left(\frac{ne}{s}\right)^s \cdot \left(\left(\frac{es}{3}\right)^{3/2}\right)^s \left(\frac{8^{3/2}k^{9/2}}{n^{3/2}}\right)^s \\ &\leq \left(\frac{54k^{9/2}s^{1/2}}{n^{1/2}}\right)^s . \end{aligned}$$

Observe that if $4 \leq s \leq \sqrt{n}$ then the last term is $O\left(\left(\frac{1}{\sqrt{n}}\right)^4\right) = o(1/n)$ and if $\sqrt{n} \leq s \leq \alpha n$, then the last term is bounded by $c^{\sqrt{n}}$ for some constant $c < 1$. Therefore in both cases the last term is $o(\frac{1}{n})$, and hence the total sum in Equation (4.2) is $o(1)$. We conclude that with high probability every induced subgraph of size αn is 3-colorable. $\square$

4.1.2. Increasing the degree. The construction in Lemma 4.1 gives a graph with constant average degree. We next consider two ways of obtaining graphs with larger degrees.

Given a graph G' over n' vertices and m' edges, a *d-blowup* of G' is a graph G obtained in the following manner. Every vertex v' of G' is transformed into a cluster of d vertices in G. Thus the number of vertices in G is $n = n'd$. Every edge of G' is transformed into d^2 edges in G that form a complete bipartite graph between the corresponding clusters, and hence the number of edges in G is $m = m'd^2$. In this the next subsection we shall use blowups to build graphs on n vertices and average degree $\Theta(d)$ that have similar properties to sparse graphs on $\Theta(\frac{n}{d})$ vertices. Clearly, if G' is k-colorable, then so is G. Moreover:

Claim 4.3. *If G' is ϵ-far from being k-colorable then G is also ϵ-far from being k-colorable.*

Proof. We call a k-coloring φ a *cluster coloring* if every two vertices from the same cluster have the same color. Clearly, every cluster coloring of G has at least ϵdn violating edges, and thus it remains to show that there is an optimal coloring (with respect to the number of violating edges) which is a cluster coloring.

To this end, consider any optimal coloring φ', and let u_1, u_2 vertices from the same cluster such that $\varphi'(u_1) \neq \varphi'(u_2)$. Denote by b_1, b_2 the number of violating edges incident to u_1, u_2 respectively. Without loss of generality assume that $b_1 \leq b_2$, then we can assign u_2 the color of u_1 without increasing the total number of violating edges. Applying this to all clusters we obtain an optimal coloring that is also a cluster coloring, and the claim follows. □

We next prove a variant of Claim 4.3 that uses a different type of blowups. We take a fixed graph H, and blow it up where between every two incident clusters we have a graph with a certain expansion property. We show that if H is not k-colorable then the obtained graph is far from being k-colorable. In particular, we have the following, where for two (disjoint) subsets of vertices A and B, we use $e(A, B)$ to denote the number of edges with one endpoint in A and one endpoint in B.

Claim 4.4. *Let $k \geq 3$ be a constant and let $H = (V, E)$ be a fixed graph that is not k-colorable. Let G be a graph with $n|H|$ vertices, obtained by transforming each vertex v in H to a cluster $U(v)$ of n vertices. Suppose that for every $(u, v) \in E(H)$, the pair $U(u), U(v)$ is transformed into a d-regular bipartite graph with the following property. There exists a constant $c(k) > 0$ such that for every $A \subseteq U(u), B \subseteq U(v)$ of size n/k each, we have $e(A, B) \geq c(k) \cdot nd$. Then G is $\Theta(1)$-far from being k-colorable.*

Proof. Fix a k-coloring of G, and identify with each cluster $U(v)$ a color i such that at least n/k of the vertices in $U(v)$ are colored by i, breaking ties arbitrarily. We get a k-coloring of H, and since H is not k-colorable, we conclude that there is an edge $(u, v) \in E(H)$ and a color i such that at least n/k vertices from $U(u)$ are colored i and at least n/k vertices from $U(v)$ are colored i. By the expansion property, there are at least $c(k) \cdot nd$ violating edges. Since there are $\Theta(nd)$ edges in G, we get that G is $\Theta(1)$-far from being k-colorable. □

4.1.3. Regular bipartite graphs. In what follows we describe some properties of (random) d-regular bipartite graphs. We denote by $\mathcal{D}_{n,d}$ the set of all d-regular bipartite graphs with two independent sets, V_1 and V_2 of size n each.

LEMMA 4.5. *Let $F = (V_1 \cup V_2, E)$ be a graph sampled uniformly from $\mathcal{D}_{n,d}$, where $d = \omega(1)$ and $d < 0.9n$, and let α be some positive constant. Then with high probability, for every two sets $X \subseteq V_1$ and $Y \subseteq V_2$ where $|X| = |Y| = \alpha n$, we have $e(X, Y) \geq 1/2 \cdot |X||Y|d/n$.*

PROOF. Consider first the binomial random bipartite graph $G(n, n, d/n)$ defined as follows. There are two sets $|V_1| = |V_2| = n$, and for every $a \in V_1, b \in V_2$, there is an edge from a to b with probability d/n. Let J^{reg} be the event where the obtained graph is d-regular, and let J^{exp} be the event that there is a pair of sets $X \subset V_1$, $Y \subset V_2$ where $|X| = |Y| = \alpha n$ and $e(X, Y) < 1/2 \cdot |X||Y|d/n$.

Observe that the obtained random graph conditioned on J^{reg} becomes a random bipartite d-regular graph from $\mathcal{D}_{n,d}$. We will argue that $\Pr[J^{\text{exp}}] = o(\Pr[J^{\text{reg}}])$, and then we conclude that almost all graphs from $\mathcal{D}_{n,d}$ satisfy the assertion of the lemma.

Observe first that by applying Chernoff's inequality together with the union bound, we get that $\Pr[J^{\text{exp}}] \leq 2^{2n} \cdot e^{-\beta dn}$ for some constant $\beta > 0$, as there are at most 2^{2n} ways to choose X, Y, and then the number of edges between X and Y is binomially distributed with probability d/n.

In order to estimate the probability of J^{reg}, we denote by J_i^{reg}, $i \in \{1, 2\}$, the event where all vertices in V_i have degree exactly d. Clearly $J^{\text{reg}} = J_1^{\text{reg}} \cap J_2^{\text{reg}}$ and $\Pr[J_1^{\text{reg}}] = \Pr[J_2^{\text{reg}}]$. Also, we have that $\Pr[Bin(n, d/n) = d] \geq \frac{c_2}{\sqrt{d}}$ for some constant c_2, assuming that $d = \omega(1)$ and $d \leq 0.9n$ (see e.g. [7, Chapter 1.2]). Finally, by a result of Ordentlich and Roth [20], the events $J_1^{\text{reg}}, J_2^{\text{reg}}$ are positively correlated. Thus we have

$$\Pr[J^{\text{reg}}] \geq \Pr[J_1^{\text{reg}}] \cdot \Pr[J_2^{\text{reg}}] \geq \left(\frac{c_2}{\sqrt{d}}\right)^{2n}.$$

We conclude that $\Pr[J^{\text{exp}}] = o(\Pr[J^{\text{reg}}])$, as claimed. □

The next lemma deals with the probability of a certain event occurring when a d-regular bipartite graph $(V_1 \cup V_2, E)$ is selected randomly from $\mathcal{D}_{n,d}$ *conditioned* on it containing a given subgraph. It will be convenient to consider vertex-labeled graphs. That is, $V_1 = \{v_1^1, \ldots, v_1^n\}$ and $V_2 = \{v_2^1, \ldots, v_2^n\}$.

Lemma 4.6. *Let $H = (V_1^H \cup V_2^H, E(H))$ be a fixed bipartite subgraph with bounded degree d, where $V_1^H \subseteq V_1$ and $V_2^H \subseteq V_2$. We denote by* $\mathrm{Ext}(H)$ *the subset of graphs $(V_1 \cup V_2, E)$ in $\mathcal{D}_{n,d}$ that contain H as a subgraph. The following holds for any $v \in V_1 \cup V_2$ and any $u, w \notin \Gamma_H(v)$ such that* $\deg_H(w) = 0$. *The probability over a uniformly selected graph $F \in$* $\mathrm{Ext}(H)$ *that $(v, u) \in E(F)$ is upper bounded by the probability that $(v, w) \in E(F)$.*

We note that the lemma holds more generally for the case that $\deg_H(u) \geq \deg_H(w)$. However, we apply it only to the case that $\deg_H(w) = 0$, and the proof of this case is slightly simpler.

Proof. First observe that the claim holds trivially if $\deg_H(u) = 0$ since in this case, the probability that $(v, u) \in E(F)$ equals the probability that $(v, w) \in E(F)$. We thus assume from this point on that $\deg_H(u) \geq 1$. The claim also holds trivially if $\deg_H(v) = d$, so we may assume that $\deg_H(v) < d$.

Assume without loss of generality that $v \in V_1$ and $u, w \in V_2$. Let $\mathrm{Ext}_{v,u}(H)$ denote the subset of graphs in $\mathrm{Ext}(H)$ in which there is an edge between v and u (recall that this edge does not belong to H), and similarly define $\mathrm{Ext}_{v,w}(H)$. We would like to show that $|\mathrm{Ext}_{v,u}(H)| \leq |\mathrm{Ext}_{v,w}(H)|$. We first note that there are graphs that belong to both sets (in case that $\deg_H(v) \leq d - 2$), and we continue by considering the sets that result from removing those graphs in the intersection. We denote the resulting sets by $\mathrm{Ext}_{v,u}^{w}(H)$ and $\mathrm{Ext}_{v,w}^{u}(H)$, respectively.

We next define an auxiliary bipartite graph whose two sides correspond to $\mathrm{Ext}_{v,u}^{w}(H)$ and $\mathrm{Ext}_{v,w}^{u}(H)$, respectively. We put an (auxiliary) edge between $F \in \mathrm{Ext}_{v,u}^{w}(H)$ and $\tilde{F} \in \mathrm{Ext}_{v,w}^{u}(H)$, if $\tilde{F}$ can be obtained from F by switching between v and some $x \in \Gamma_F(w) \setminus \Gamma_F(u)$. That is, F and $\tilde{F}$ are the same, except that in F we have the edges (v, u) and (x, w) (that do not appear in $\tilde{F}$), while in $\tilde{F}$ we have the edges (v, w) and (x, u) (that do not appear in F).

Suppose we partition the graphs on each side of the auxiliary bipartite graph into subsets according to the size of $|\Gamma_F(u) \cap \Gamma_F(w)|$ (which ranges between 0 and $d - 1$). Then there are edges only between graphs in which this intersection has the same size (and is in fact equivalent). Since the degree of each $F \in \mathrm{Ext}_{v,u}^{w}(H)$ equals $d - |\Gamma_F(u) \cap \Gamma_F(w)|$, while the degree of each $F \in \mathrm{Ext}_{v,w}^{u}(H)$ is at most $d - |\Gamma_F(u) \cap \Gamma_F(w)|$ (in fact, is strictly smaller), we get that $|\mathrm{Ext}_{v,w}^{u}(H)| \geq |\mathrm{Ext}_{v,u}^{w}(H)|$, so that $|\mathrm{Ext}_{v,w}(H)| \geq |\mathrm{Ext}_{v,u}(H)|$, as claimed. □

4.2. Lower Bounds for Group Queries. In this section we prove lower bounds for the group query model, and since we have proved that the group query model is essentially at least as strong as the combined model (see Claim 2.1), our results apply also to the latter model.

We start with a simple consequence of the construction of Lemma 4.1 to get a one-sided error lower bound, and then use a more complex argument to get a two-sided error bound.

COROLLARY 4.7. *Every one-sided error algorithm for testing k-colorability, $k \geq 3$, must perform $\Omega\left(\frac{n}{d}\right)$ group queries.*

PROOF. Every one-sided error algorithm must accept whenever the subgraph it has queried is k-colorable. Take a graph over $\Theta(\frac{n}{d})$ vertices that is obtained as described in Lemma 4.1, and blow it up by factor d. By Claim 4.3, with high probability the obtained graph is $\Theta(1)$-far from being k-colorable. Moreover, every induced subgraph of size $\alpha \cdot \frac{n}{d}$ is k-colorable, where α is an absolute constant as in Lemma 4.1, and thus in order to find a $(k+1)$-critical subgraph we need to get at least $\alpha \cdot \frac{n}{d}$ positive answers. Every query returns at most one positive answer, and therefore we need to perform at least that number of queries in order to obtain evidence that the graph is not k-colorable. The corollary follows. $\square$

Next we build on a reduction from [6] to get a lower bound of $\Omega\left(\frac{n}{d}\right)$ for two-sided error algorithms in the group query model. The lower bound in [6] is for the case $k = 3$, and therefore we start with a reduction for general k.

LEMMA 4.8. *For any n and constant $k \geq 4$ there exists a constant degree graph G' on $O(n)$ vertices with a set $S \subseteq V(G')$ of size n such that the following holds. For every graph F on n vertices with constant maximum and average degrees, if we replace $G'[S]$ by an induced copy of F (while keeping all edges between S and $V(G') \setminus S$), and we denote the resulting graph by G'', then we have:*

- *If F is 3-colorable then G'' is k-colorable.*
- *If F is $\Theta(1)$-far from being 3-colorable then G'' is $\Theta(1)$-far from being k-colorable*

PROOF. We show that a graph G' chosen from a certain distribution satisfies the properties with high probability, and it will follow that such a graph exists. Consider first a random graph H_0 from the following distribution of graphs on $(k-3)n$ vertices. The vertices are partitioned into $k-3$ sets $U_1, U_2, \ldots, U_{k-3}$ of equal size. Every two vertices from different clusters are connected by an edge independently with probability $p = \frac{ck \log k}{n}$ for some constant c, and every cluster forms an independent set of size n. Clearly, every graph from such a distribution is $(k-3)$-colorable. We will show that with high probability it is $\Theta(1)$-far from being $(k-4)$-colorable. First, with high probability such a graph has at most ck^4n edges. Consider a coloring χ of the vertices of H_0 with $k-4$ colors. Denote by n_i, $1 \le i \le k-4$, the number of vertices that are colored i. Let $y_i = \max\{n_i - n, 0\}$ and observe that $\sum_{i=1}^{k-4} y_i \ge n$. Call a pair of vertices *dangerous* with respect to χ if they are from different clusters but have the same color in χ. For every color class with $n + y_i$ vertices, there are at least $n \cdot y_i$ dangerous pairs. Hence, the total number of dangerous pairs is at least n^2. Therefore, by Chernoff-type bounds for every fixed coloring, the probability that there are less than $n/(4k)$ violating edges is bounded by $e^{-\Theta(pn^2)} = o(k^{-(k-3)n})$ for sufficiently large n. Using the union bound, we get that almost always a graph from this distribution is $\Theta(1)$-far from being $(k-4)$-colorable.

We next establish that with high probability H_0 is close to a graph with degrees bounded by a constant (which is still $\Theta(1)$-far from being $(k-4)$-colorable). Let $t = ne^{-np}$. Let A be a subset of t vertices in H_0. Observe that

$$\sum_{v \in A} d_{H_0}(v) = 2e_{H_0}(A) + e_{H_0}(A, V \setminus A),$$

where $e_{H_0}(A)$ is the number of edges of H_0 within A and $e_{H_0}(A, V \setminus A)$ is the number of edges of H_0 with one endpoint in A and another outside A. Hence if $\sum_{v \in A} d_{H_0}(v) \ge 10(k-3)np \cdot t$, we get that $e_{H_0}(A) > (k-3)npt$ or $e_{H_0}(A, V \setminus A) > 8(k-3)npt$. The probability of the former can be bounded using the union bound by

$$\begin{aligned} &\binom{(k-3)n}{t} \cdot \binom{\binom{t}{2}}{(k-3)npt} \cdot p^{(k-3)npt} \\ &\le \left(\frac{e(k-3)n}{t} \cdot \left(\frac{et}{2(k-3)n} \right)^{(k-3)np} \right)^t = o(1)\,. \end{aligned}$$

The probability of the latter is bounded by

$$\begin{aligned} \binom{n(k-3)}{t} \binom{t(n(k-3)-t)}{8n(k-3)pt} p^{8n(k-3)pt} &\le \left(\frac{en(k-3)e^{8n(k-3)p}}{t \cdot 8^{8n(k-3)p}} \right)^t \\ &= \left(3k\, e^{np} \left(\frac{e}{8} \right)^{8(k-3)np} \right)^t = o(1)\,. \end{aligned}$$

Assume none of the above two events holds for any subset of t vertices in H_0, and choose A to be a subset of t vertices of largest degrees in H_0, breaking ties arbitrarily. Then $\sum_{v\in A} d_{H_0}(v) \leq 10(k-3)np\cdot t \leq nk^{-\Theta(k^2)}$. It follows also that the vertex degrees outside A are all at most $10(k-3)np$ in H_0. Delete all edges of H_0 touching A and denote the obtained graph by H; the number of deleted edges is at most $nk^{-\Theta(k^2)}$. Then H is (1) $(k-3)$-colorable, (2) $\Theta(1)$-far from being $(k-4)$-colorable and (3) both its maximum degree and its average degree are $\Theta(1)$.

We now define a graph G' over $O(n)$ vertices, and a subset $S \subset V(G')$ of size n as required in the lemma (recall that the claim in the lemma is about replacing $G'[S]$ by an induced copy of a graph F). G' consists of a copy of H, and the set S, which is an independent set disjoint of $V(H)$, where the edges between $V(H)$ and S are defined as follows. Let $c_1 = c_1(k)$ be a constant that will be set subsequently. Every pair of vertices $v_1 \in V(H)$ and $v_2 \in S$ are connected with probability $\frac{c_2}{n}$ for a constant c_2, where c_2 is selected so that the following holds with high probability: For every choice of subsets $X \subset V(H)$ and $Y \subset S$ such that both have size n/c_1, there are at least $c_3 n$ edges between X and Y for some constant c_3. Assume from this point on that this is in fact true. By applying essentially the same argument used to bound the degrees of vertices in H, it is possible to remove a very small number of edges between $V(H)$ and S so as to ensure that all degrees are upper bounded by a constant (while still maintaining that there are $\Omega(n)$ edges between every pair X and Y as defined above).

For any fixed choice of a graph F over n vertices (with constant maximum and average degrees), let G'' be the graph that results from replacing $G'[S]$ with F. Since H is $(k-3)$-colorable, if F is 3-colorable then G'' is k-colorable, as required. We thus turn to the case that F is ϵ-far from being 3-colorable for $\epsilon = \Theta(1)$. For an illustration of the cases considered in the argument presented next, see Figure 4.1.

Consider any coloring χ of $V(G'')$ with k colors, and let $C_1, \ldots, C_k$ be the corresponding color classes. Recall that H is ϵ'-far from being $(k-4)$-colorable, for $\epsilon' = \Theta(1)$. Denote $C_i \cap V(H)$ by C_i^H and $C_i \cap V(F)$ by C_i^F. Let r^H be the ratio between the maximum degree in H and the average degree, and define r^F analogously. Recall that both r^H and r^F are constants. Let $\epsilon'' = \frac{1}{2k\max\{r^H,r^F\}}\min\{\epsilon,\epsilon'\}$, and suppose that there exists a color class C_i such that $|C_i^H| \geq \epsilon'' n$ and $|C_i^F| \geq \epsilon'' n$. In such a case, by the foregoing discussion, setting $c_1 = 1/\epsilon''$, there are $\Omega(n)$ edges between C_i^H and C_i^F, and so there is a constant fraction of violating edges with respect to the coloring χ.

Otherwise, for each C_i, let C_i' be the larger between C_i^H and C_i^F, and observe that $|\bigcup_i (C_i \setminus C_i')| \leq \frac{1}{2\max\{r^H,r^F\}}\min\{\epsilon,\epsilon'\}n$. First consider the case that $V(F)$ contains at most 3 subsets $C_{i_1}', C_{i_2}', C_{i_3}'$ (and hence it contains

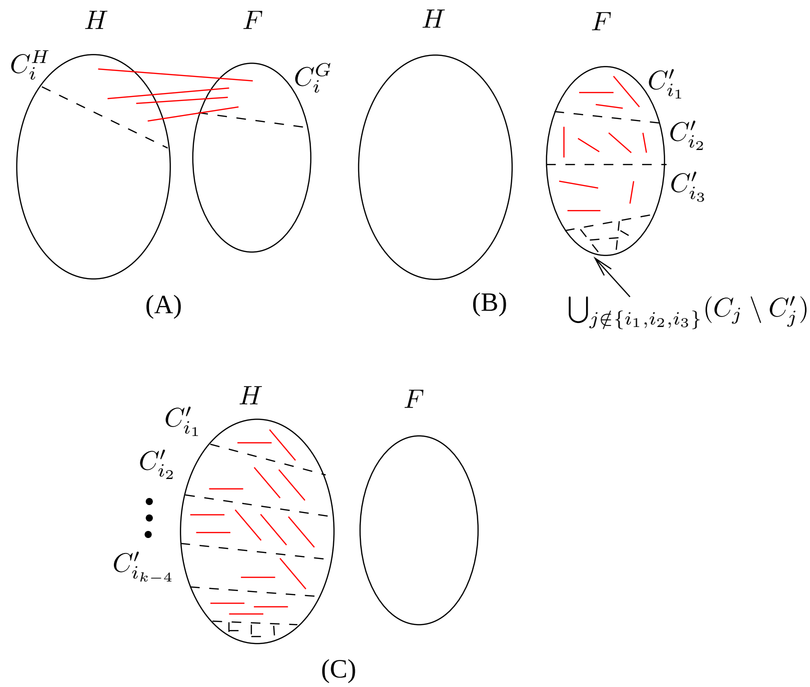

Figure 4.1: An illustration for the different cases of the argument showing that G'' is $\Omega(1)$-far from k-colorable in the proof of Lemma 4.8. In Case (A) there exists a color class C_i such that $|C_i^H| \geq \epsilon'' n$ and $|C_i^F| \geq \epsilon'' n$. In Case (B) there are at most three large subsets $C'_{i_1}, C'_{i_2}, C'_{i_3}$ in $V(F)$. In Case (C) there are more than three such subsets, and so there are at most $k-4$ subsets $C'_{i_1}, \ldots, C'_{i_{k-4}}$ in $V(H)$.

$\bigcup_{i \notin \{i_1,i_2,i_3\}}(C_i \setminus C'_i))$. We claim that since F is ϵ-far from 3-colorable, there must be $\Omega(n)$ violating edges with respect to χ. To verify this consider any three-way partition (V_1, V_2, V_3) of $V(F)$ such that $C'_{i_j} \subseteq V_j$ for $j \in \{1,2,3\}$. Since F is ϵ-far from 3-colorable there are at least $\epsilon d^F n$ violating edges with respect to this partition, where d^F is the average degree in F. However, the number of edges incident to vertices in $\bigcup_{i \notin \{i_1,i_2,i_3\}}(C_i \setminus C'_i)$ is at most $\frac{1}{2r^F}\epsilon d^F_{\max} n = \frac{1}{2}\epsilon d^F n$, where $d^F_{\max}$ is the maximum degree in F. Therefore, there are at least $\frac{1}{2}\epsilon d^F n = \Omega(n)$ violating edges between pairs of vertices that belong to the same C'_{i_j}.

Finally, if $V(F)$ contains more than 3 such subsets, then $V(H)$ must contain at most $k-4$ such subsets. Since H is ϵ'-far from being $(k-4)$-colorable, in this case too there must be a constant fraction of violating edges with respect to χ. $\square$

Using Lemma 4.8, we can now prove the lower bound on group queries:

PROOF OF ITEM 2 IN THEOREM 1.3. We first prove the bound for $k = 3$, and in the end of the proof we explain how to adjust it to other values of k. The authors of [6] show that there exist two distributions of b-regular graphs (for a constant b) over $\tilde{n}$ vertices with the following properties. All graphs in the support of the first distribution, denoted $\widetilde{D}_{\text{3col}}$, are 3-colorable, and with high probability over the choice of a graph according to the second distribution, denoted $\widetilde{D}_{\text{3col-far}}$, the graph is $\Theta(1)$-far from being 3-colorable. However, the distributions are statistically indistinguishable by a tester that performs $o(\tilde{n})$ neighbor queries.

We first observe that the distributions $\widetilde{D}_{\text{3col}}$ and $\widetilde{D}_{\text{3col-far}}$ are also statistically indistinguishable by any tester that performs $o(\tilde{n})$ group queries. Assume, in contradiction, that there exists a tester that performs $q(\tilde{n}) = o(\tilde{n})$ groups queries and distinguishes with high constant probability between a graph selected randomly according to $\widetilde{D}_{\text{3col}}$ and a graph selected randomly according to $\widetilde{D}_{\text{3col-far}}$. But then, since the graphs considered are b-regular, where b is a constant, it is possible to answer each group query by performing b neighbor queries, and thus to distinguish between the two distributions by performing $b \cdot q(\tilde{n}) = o(\tilde{n})$ neighbor queries. From this point on, when we discuss testers, we mean testers that perform group queries (and possibly degree queries, though these are not useful since the graphs are regular).

We next define two distributions, $D_{\text{3col}}, D_{\text{3col-far}}$ that are created by d-blowups of the graphs in the supports of the original distributions $\widetilde{D}_{\text{3col}}, \widetilde{D}_{\text{3col-far}}$, respectively. Given a graph $\widetilde{G}$ on $\tilde{n}$ vertices, let G denote the d-blowup of $\widetilde{G}$. Clearly, all graphs in the support of D_{3col} are 3-colorable, while by Claim 4.3, for every graph G in the support of $\widetilde{D}_{\text{3col-far}}$ that is $\Theta(1)$-far from being 3-colorable, we also have that $\widetilde{G}$ (in the support of $D_{\text{3col-far}}$) is $\Theta(1)$-far from being 3-colorable. Suppose that there exists a tester T that performs $o(n/d)$ group queries when given access to a graph with average degree d. In particular, this tester distinguishes (with high success probability) between a random graph selected according to $D_{\text{3col-far}}$ and a random graph selected according to D_{3col} by performing $o(n/d) = o(\tilde{n})$ group queries to the graph. We define the following tester $\widetilde{T}$ that distinguishes between a random graph from $\widetilde{D}_{\text{3col}}$ and a random graph from $\widetilde{D}_{\text{3col-far}}$ by performing $o(\tilde{n})$ group queries. By the definition of a d-blowup, every group query to G can be emulated using a single group query to $\widetilde{G}$. Thus, $\widetilde{T}$ is simply defined by (virtually) running T on G, answering each group query it asks by performing a corresponding query to $\widetilde{G}$, and outputting the answer that T outputs.

Turning to the case that $k > 3$, we can adjust the proof as follows. We replace the distributions $\widetilde{D}_{\text{3col}}$ and $\widetilde{D}_{\text{3col-far}}$ by two new distributions $\widetilde{D}_{\text{k-col}}$ and $\widetilde{D}_{\text{k-col-far}}$. Every graph F from the distributions $\widetilde{D}_{\text{3col}}$ and $\widetilde{D}_{\text{3col-far}}$ (which is a regular graph) is replaced by a graph G'' given by Lemma 4.8. The new graph has $O(\tilde{n})$ vertices and degrees bounded by a constant. If the graph G'' is generated based on a graph from $\widetilde{D}_{\text{3col}}$ (i.e., according to the induced distribution $\widetilde{D}_{\text{k-col}}$), then it is k-colorable and otherwise it is $\Theta(1)$-far from being k-colorable (with high probability). We may assume that the tester knows in advance all the edges in G'' except the edges in the induced copy of F. Since the graph G'' is fixed (except for the induced copy of F), the only way to separate between the two resulting distributions is through the induced copy of F. Finally, we use a d-blowup (again, we can assume that the tester knows in advance the clusters partition). Following the same argument as for the case of $k = 3$, we get a lower bound of $\Omega\left(\frac{n}{d}\right)$ for every value of k, as required. $\square$

4.3. Lower Bounds for Neighbor Queries. Our lower bound proof for neighbor queries will use directed orientations of undirected graphs. For a fixed orientation O, the indegree of a vertex v is the number of edges that are directed into v. We start with two simple claims.

Claim 4.9. *Let H be a graph over n vertices with average degree d, and consider any orientation of H. Then there exists a vertex with indegree at least $\lceil d/2 \rceil$.*

Proof. The total number of edges is the sum of the indegrees taken over all vertices in the graph. If all the vertices have indegree less than $\lceil d/2 \rceil$ then the total number of edges is less than $\frac{nd}{2}$ which contradicts the assumption. $\square$

Claim 4.10. *Let H be a K_4-free graph that is not k-colorable for $k \geq 6$, and let O be an orientation of H. The set of vertices with indegree at least $\lceil \frac{k}{2} \rceil - 1$ with respect to O induces a non-3-colorable graph.*

Proof. Denote the set of vertices with indegree at least $\lceil k/2 \rceil - 1$ by C and the complement of C by U. In U every vertex has indegree at most $\lceil k/2 \rceil - 2$. Suppose that $H[U]$ is not $(k-3)$-colorable. Then $H[U]$ must contain a $(k-2)$-critical subgraph. Since that subgraph doesn't contain a copy of K_4, by Brooks'

Theorem (see e.g., [25, Thm. 5.1.22]), it has average degree more than $k-3$. By Claim 4.9, such a subgraph must contain a vertex with indegree more than $\lceil (k-3)/2 \rceil$; the latter quantity is at least $\lceil k/2 \rceil - 1$, and we have reached a contradiction. If $H[C]$ is 3-colorable, we get that the whole graph can be colored with k colors. This contradicts the premise of the lemma (that H is not k-colorable), and thus the proof is completed. □

We first prove a two-sided error bound that follows immediately from previous results.

Proof of Item 2 in Theorem 1.2. As the group query model is at least as strong as the neighbor query model (Claim 2.1), by Theorem 1.3 we get a lower bound of $\Omega\left(\frac{n}{d}\right)$ queries. In [18] it was proved that every tester for k-colorability (possibly with two-sided error) that performs only neighbor queries requires $\Omega(\sqrt{n})$ queries. The claim follows. □

Next we prove a uniform bound valid for every value of d.

Proof of Item 3 in Theorem 1.2. Let H be a fixed $(k+1)$-critical graph that is not K_{k+1}, where we assume that H is known to the tester. (In the course of the proof we shall make several such assumptions that can only make the task of the tester easier.) Based on H, we construct a graph G while interacting with the tester. Initially, the graph G contains a cluster of $n' = \frac{n}{|H|}$ vertices for each vertex in H, where the tester knows for each vertex in G to which cluster it belongs. We say that two clusters are *incident*, if the corresponding vertices in H are neighbors. In the course of the execution of the tester we construct a random regular bipartite graph between every two incident clusters, and when the execution of the tester ends, we complete every incident pair of clusters to a (random) d-regular bipartite graph.

Recall that the tester has to find evidence that the graph is not k-colorable. Observe first that the tester knows the degrees of all vertices, and so we may focus only on the neighbor queries it makes. Suppose that it makes a neighbor query from a vertex v in a cluster U_1. We may assume that the tester can request a neighbor from an incident cluster of its choice. (Recall that in each cluster there are $\Theta(n)$ vertices.) Suppose that the tester requests a neighbor from U_2. In order to answer the query, we consider all graphs in $\mathcal{D}_{n',d}$ that agree with the current knowledge graph between the clusters U_1, U_2, and return as an answer a vertex in U_2 that is selected at random according to the distribution induced on the neighbors of v conditioned on this knowledge graph. At the end of the execution of the tester we choose randomly and uniformly for every pair of clusters a d-regular bipartite graph from $\mathcal{D}_{n',d}$ that is consistent with the final knowledge graph.

By the definition of the above process, for every tester, between each pair of incident clusters we have a random d-regular bipartite graph. By Lemma 4.5 and Claim 4.4, we get that with high probability, the obtained graph is $\Theta(1)$-far from being k-colorable.

We also claim that as long as the tester performs $o(n)$ queries, the probability that a query is answered by a vertex that already belongs to the knowledge graph is upper bounded by $O(1/n')$. To this end, we apply Lemma 4.6. Consider a request for a neighbor of $v \in U_1$ that belongs to U_2. By the lemma, for every fixed vertex $u \in U_2$ and any vertex w that is an isolated vertex in the knowledge graph, the probability of getting u as an answer is upper bounded by the probability of getting w as an answer. Since there are $o(n')$ non-isolated vertices, we get that the probability to get any fixed vertex is bounded by, say, $2/n'$.

In order to complete the proof, we show that with high probability, the tester does not reveal a non k-colorable graph throughout its execution.

During the execution of the tester, we represent the knowledge graph as a directed graph, where if the tester performs a neighbor query from a vertex v and the answer is u, then we orient the edge from v to u. As every minimal evidence is a $(k+1)$-critical graph that is not a clique, by Brooks' Theorem (see e.g., [25, Thm. 5.1.22]) it has average degree more than k. By Claim 4.9 it has at least one vertex with indegree at least $\lceil (k+1)/2 \rceil$.

We now wish to prove that in order to find a single vertex in the graph with such indegree, the tester needs to perform many queries. Every neighbor query return as an answer each particular vertex with probability at most $2/n'$. Therefore, the probability that after r queries, some vertex was selected more than $\lceil (k+1)/2 \rceil$ times is at most

$$\binom{r}{\lceil (k+1)/2 \rceil} \cdot \left(\frac{2}{n'}\right)^{\lceil (k+1)/2 \rceil} \leq \left(\frac{2r}{n'}\right)^{\lceil (k+1)/2 \rceil} .$$

Applying the union bound, if $r = O(n^{1-\frac{1}{\lceil (k+1)/2 \rceil}})$, then with high probability no vertex is selected more than $\lceil (k+1)/2 \rceil$ times, and so the bound on the number of needed queries follows.

Proof of Item 4 in Theorem 1.2. By Lemma 4.1 there exists a graph over $\Theta(\frac{n}{d})$ vertices that is $\Theta(1)$-far from being k-colorable, yet every induced subgraph of size $\alpha \frac{n}{d}$ (for some constant $\alpha > 0$) is 3-colorable (in particular, the graph does not contain K_4 as a subgraph). Let G be a d-blowup of such a graph. By Claim 4.3, G is also K_4-free and $\Theta(1)$-far from being k-colorable.

As in the proof of Item 3 in the theorem, during the execution of the tester, we represent the knowledge graph as a directed graph, where if the tester performs a neighbor query from a vertex v and the answer is u, then we orient the edge from v to u. Denote the directed knowledge graph by H. If H can serve as evidence that the graph G is not k-colorable, then H contains a $(k+1)$-critical subgraph. By Claim 4.10, the set of vertices in H with indegree at least $\lceil k/2 \rceil - 1$ induces a subgraph that is not 3-colorable, and therefore by the properties of the construction, this subgraph has vertices with indegree at least $\lceil k/2 \rceil - 1$ from $\Theta(\frac{n}{d})$ distinct clusters.

We next lower bound the number of queries required in order to find a single vertex with indegree at least $\lceil k/2 \rceil - 1$. Suppose that the tester makes a neighbor query from a vertex u that belongs to a cluster U_1, and obtains as an answer a vertex from a cluster U_2. If the tester makes $o(n)$ queries, for all but $o(n/d)$ of the clusters, it gets at most d/c answers within the cluster (for some constant c). For each such cluster, the probability of getting any particular vertex as an answer in a certain query is at most c/d. Therefore, among r queries that return neighbors in a certain cluster, the probability of getting the same vertex as an answer $\lceil k/2 \rceil - 1$ times is bounded by

$$d \cdot \binom{r}{\lceil k/2 \rceil - 1} \cdot \left(\frac{c}{d}\right)^{\lceil k/2 \rceil - 1} \leq d \cdot \left(\frac{cr}{d}\right)^{\lceil k/2 \rceil - 1}.$$

It follows that with high probability, $\Omega\left(d^{1-\frac{1}{\lceil k/2 \rceil - 1}}\right)$ neighbor queries are needed to find a vertex with degree $\lceil k/2 \rceil - 1$ that belongs to a certain cluster. Recall that the tester has to find such vertices from $\Theta(\frac{n}{d})$ distinct clusters, and therefore every one-sided error tester must perform

$$\Theta\left(\frac{n}{d}\right) \cdot \Omega\left(d^{1-\frac{1}{\lceil k/2 \rceil - 1}}\right) = \Omega\left(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}}\right)$$

queries. The theorem follows.

4.4. On the power of the combined model. Let $\mathbb{T}$ be the distribution of the d-blowups of graphs given by Lemma 4.1. We shall use the notation $G \in \mathbb{T}$ to mean a graph G selected according to the distribution $\mathbb{T}$. Observe that with high probability over the choice of $G \in \mathbb{T}$, the average degree in G is $\Theta(d)$. We have seen that for this distribution, there exist lower bounds for the pair query and neighbor query models of $\Omega\left(\left(\frac{n}{d}\right)^2\right)$ and $\Omega\left(n \cdot d^{-\frac{1}{\lceil k/2 \rceil - 1}}\right)$ (if $k \geq 6$) respectively. Here we describe a one-sided error algorithm in the combined model that uses $\tilde{O}(\frac{n}{\sqrt{d}})$ queries for this distribution. This shows that the combined model is stronger than the optimum of the pair query and neighbor query models for the problem of k-colorability.

PROPOSITION 4.11. *There exists an algorithm such that, given a graph $G \in \mathbb{T}$, reveals a non k-colorable subgraph using $\tilde{O}(\frac{n}{\sqrt{d}})$ queries in the combined model with high probability (the probability is taken both over the choice of the graphs and the success probability of the algorithm).*

PROOF SKETCH. Let $G \in \mathbb{T}$ and let W be a set of vertices in G. For a vertex v with a non-empty neighborhood we can create an *identifier* $I(v)$ by taking $O(\sqrt{d}\log n)$ random neighbors. By the Birthday Paradox, with high probability every two vertices $v_1, v_2 \in W$ from the same cluster will have a common element in their identifiers.

Given two vertices u and v, we can also check if they have the same set of neighbors in the following way. We query $O(\log n)$ random neighbors from u, and perform a pair query for every such neighbor to v. We then query $O(\log n)$ random neighbors of v, and perform a pair query for every such neighbor to v. If u and v are not in the same cluster (and their clusters have distinct sets of neighbors), then with high probability we find a vertex r connected only to one of them. We call this procedure the *explicit test*. Note that this procedure cannot distinguish between the case where two vertices are from the same cluster, and the case where they belong to distinct clusters but have the same set of neighbors. However, this is immaterial to the argument, and distinguishing between the two cases is not necessary (and is impossible).

Our algorithm works as follows. We first take a sample S of size $O\left(\frac{n \log n}{d}\right)$ vertices from G. By the known bounds of the Coupon Collector's Problem (see [19, Thm. 3.8]), with high probability we have a vertex from every cluster in the graph. We then build an identifier for every vertex in S. For every pair of vertices that have a common neighbor in their identifiers, we apply the explicit test. For every vertex there are $O(\log^2 n)$ clusters of vertices within distance at most 2 from it. Therefore, with high probability we apply the explicit test $O(\log^3 n)$ times for every vertex from S, and the total number of queries in this step is $\tilde{O}(\frac{n}{\sqrt{d}})$. After this step we can group the vertices of S to clusters. Let $S' \subseteq S$ be a set of vertices with a single vertex from every cluster.

In the second step we want to check which clusters are connected. This step follows similar lines as before. For every vertex $u \in S'$ we take a random sample $R(u)$ of size $O(\log n)$. We again create identifiers for every vertex in $R(u)$, and by comparing their identifiers to those we create in the previous step, we can identify the cluster of every vertex in $R(u)$ using $\tilde{O}(\frac{n}{\sqrt{d}})$ pair and neighbor queries.

In the third step we can recover the underlying clusters graph G', since we know a vertex from every cluster, and the clusters adjacent for every such vertex. In order to get a one-sided error algorithm, we can now use pair queries in order to verify that indeed G' is a subgraph of G. This step takes $\tilde{O}(\frac{n}{d})$ queries, and so with high probability we find evidence that G is not k-colorable, as claimed. $\square$

5. Concluding Remarks and Open Questions

Our upper and lower bounds are nearly tight for the problem of one-sided error testing of k-colorability in the pair query model and in the neighbor query model (for large enough k). In the group query model our bounds are tight even for two-sided error testers. It seems plausible that the correct lower bounds for the two-sided case in the pair query model and in the neighbor query model are $\Omega\left(\left(\frac{n}{d}\right)^2\right)$ and $\Omega(n)$, respectively.

The exact power of the combined model for testing k-colorability is still not known. It seems that for the problem of testing bipartiteness, the group query model is strictly stronger than the combined model, as one can sample an induced sparse subgraph more efficiently using this query. On the other hand, we provided a distribution for which the combined model is strictly stronger than the optimum of the pair and neighbor models. It would be interesting to give tight bounds for the problem of testing k-colorability in this model.

Finally, the exact query complexity of various models is still not determined for many problems, including problems that have already been studied in the combined model. For example, it may be interesting to determine the query complexity of testing bipartiteness in the pair query model.

Acknowledgements

The work of Michael Krivelevich was supported in part by a USA-Israel BSF grant and by a grant from the Israel Science Foundation, while the work of Dana Ron was supported in part by the Israel Science Foundation (grants No. 89/05 and 246/08). We would like to thank Elad Verbin for helpful discussions that led to better results on sampling edges. We would also like to thank the reviewers of this paper for many helpful comments.

References

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

[2] N. Alon, E. Fischer, I. Newman, and A. Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. In *Proceedings of the 38th ACM STOC*, pages 251–260, 2007.

[3] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron. Testing triangle freeness in general graphs. In *Proceedings of the 17th ACM-SIAM SODA*, pages 279–288, 2006.

[4] N. Alon and M. Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Math*, 15(2):211–227, 2002.

[5] N. Alon and J. Spencer. *The probablistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[6] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability in bounded degree graphs. In *Proceedings of the 43th IEEE FOCS*, pages 93–102, 2002.

[7] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001. 2nd edition.

[8] A. Czumaj, A. Shapira, and C. Sohler. Testing hereditary properties of non-expanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6):2499–2510, 2009. This is a journal version of [9].

[9] A. Czumaj and C. Sohler. On testable properties in bounded degree graphs. In *Proceedings of the 18th ACM-SIAM SODA*, pages 494–501, 2007.

[10] D. Du and F. Hwang. *Combinatorial group testing and its applications*. World Scientific, 1993.

[11] P. Erdős. On circuits and subgraphs of chromatic graphs. *Mathematika*, 9:170–175, 1962.

[12] U. 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.

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

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

[15] O. Goldreich and D. Ron. A sublinear bipartite tester for bounded degree graphs. *Combinatorica*, 19(3):335–373, 1999.

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

[17] O. Goldreich and D. Ron. Approximating average parameters of graphs. *Random Structures and Algorithms*, 32(4):473–493, 2008.

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

[19] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

[20] E. Ordentlich and R. Roth. Two-dimensional weight-constrained codes through enumeration bounds. *IEEE Trans. Inf. Theory*, 46:1292–1301, 2000.

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

[22] V. Rödl and R. Duke. On graphs with small subgraphs of large chromatic number. *Graphs Combin*, 1:91–96, 1985.

[23] D. Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5, 2010.

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

[25] D. West. *Introduction to Graph Theory*. Prentice Hall, 2000.

Ido Ben-Eliezer
School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
idobene@post.tau.ac.il

Tali Kaufman
Bar-Ilan University and Weizmann Institute, Israel.
kaufmant@ias.edu.

Michael Krivelevich
School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel.
krivelev@post.tau.ac.il.

Dana Ron
School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel.
danar@eng.tau.ac.il.