A Nearly-Quadratic Gap Between Adaptive and Non-Adaptive Property Testers

Thesis by

Jeremy Hurwitz

In Partial Fulfillment of the Requirements
for the Degree of
Master Of Science

California Institute of Technology
Pasadena, California

2011

(Submitted November 18, 2011)

To my friends and family.

Acknowledgements

I would first like to thank my adviser, Prof. Chris Umans, for many helpful discussions throughout this entire process, from initially finding a project through formal publication and writing this thesis. I would like to also thank Prof. Oded Goldreich for encouraging me to extend the lower-bound into a hierarchy theorem.

I would also like to thank my adviser, Prof. Leonard Schulman, for supporting my frequently-wandering interests.

Abstract

We show that for all integers $t \geq 8$ and arbitrarily small $\epsilon > 0$, there exists a graph property Π (which depends on ϵ) such that ϵ-testing Π has non-adaptive query complexity $Q = \widetilde{\Theta}(q^{2-2/t})$, where $q = \widetilde{O}(\epsilon^{-1})$ is the adaptive query complexity. This resolves the question of how beneficial adaptivity is, in the context of proximity-dependent properties ([GR07]). This also gives evidence that the canonical transformation of Goldreich and Trevisan ([GT03]) is essentially optimal when converting an adaptive property tester into a non-adaptive property tester.

To do so, we consider the property of being decomposable into a disjoint union of subgraphs, each of which is a (possibly unbalanced) blow-up of a given base-graph H. In [GR09], Goldreich and Ron proved that when H is a simple t-cycle, the non-adaptive query complexity is $\Omega(\epsilon^{-2+2/t})$, even under the promise that G has maximum degree $O(\epsilon N)$. In this thesis, we prove a matching upper bound for the non-adaptive complexity and a tight (up to a polylogarithmic factor) upper bound on the adaptive complexity.

Specifically, we show that for all H, testing whether G is a collection of blow-ups of H and has maximum degree $O(\epsilon N)$ requires only $O(\epsilon^{-1} \lg^3 \epsilon^{-1})$ adaptive queries or $O(\epsilon^{-2+1/(\delta+2)} + \epsilon^{-2+2/W})$ non-adaptive queries, where $\delta = \Delta(H)$ is the maximum degree of H and $W < |H|^2$ is a bound on the size of witnesses against H.

Contents

Chapter 1

Introduction

1.1 Sublinear Algorithms and Property Testing

In recent decades, the amount of data being generated and processed has grown exponentially. As a result, in many situations “polynomial-time” is no longer a sufficient criteria for an algorithm to be considered efficient. Indeed, even quasi-linear time can be prohibitively slow for many datasets. In other situations, such as biology or sociology, the total amount of data may be only moderately large, but obtaining a single piece of data requires running an expensive experiment.

Sublinear algorithms attempt to circumvent these difficulties by considering only a minuscule random subset of the data, in the hopes of getting an approximately-correct answer far faster than would be required to obtain the exact answer. A classic example of this tradeoff is determining the mean of a list of numbers. Calculating the mean exactly requires linear time. However, by randomly sampling from the list $O(\epsilon^{-2})$ times, we can obtain an answer that is accurate to within a multiplicative factor of $1 + \epsilon$ with high probability. This observation provides the statistical justification to most population surveys. More modern examples include estimating the weight of the minimum spanning tree of a graph ([CRT05]) and estimating the distance from uniform of a given distribution ([GR00]).

In addition to estimating large-scale parameters of the input, we also might be interested in determining structural properties for the data. For example, given a list of numbers, we may wish to determine whether the list is sorted. As in the case of calculating the mean, determining the answer with certainty requires inspecting the entire input. However, unlike when approximating the mean, the answer in this case is boolean and so the idea of approximating the answer no longer makes sense. We overcome this by moving the approximation from the output to the input. Originally formalized in [BLR90] and [RS96], a *property tester* attempts to differentiate inputs which satisfy the property (the list is sorted) from inputs which are far from satisfying the property (at least an ϵ-fraction of the elements are out of place). For inputs which are close to satisfying the property, either answer is allowed – technically, the list is not sorted, but for all practical purposes it might as well be.

The formal definition of a property tester is as follows. Let $\mathcal{F}_N$ be the set of functions $[N] \to \mathcal{R}$, for some range $\mathcal{R}$. A *property* Π is an arbitrary subset of $\bigcup_{N=1}^{\infty} \mathcal{F}_N$. We say that a function $f : [N] \to \mathcal{R}$ is *ϵ-far* from Π if, for any $g \in \Pi$, $|\{x \in [N] : f(x) \neq g(x)\}| > \epsilon N$.

For example, viewing $f \in \mathcal{F}_N$ as defining a list, we can let Π be the set of sorted lists. Then a function f is ϵ-far from Π if at least ϵN elements are incorrectly sorted.

Definition 1 (Property Tester)**.** *An algorithm $\mathcal{A}$ is a* property tester *for Π if, given $\epsilon > 0$ and query access to f,*

- $\Pr[\mathcal{A}(f) = 1] > \frac{2}{3}$ *for all $f \in \Pi$*
- $\Pr[\mathcal{A}(f) = 1] < \frac{2}{3}$ *for all f which are ϵ-far from Π.*

The tester is considered *efficient* if the query-complexity $Q = Q(N, \epsilon)$ is a function of only ϵ. If $\Pr[\mathcal{A}(f) = 1] = 1$ for all $f \in \Pi$, the tester is said to have *one-sided error*; otherwise, it has two-sided error.

This framework was first applied to combinatorial structures in [GGR98], specifically with respect to graph properties. A graph property is a property which is preserved under permutation of the underlying graph (ie, the property is of the graph itself, and not of the particular labeling of the graph). In this context, graphs are generally represented using one of three different models, according to the "standard" edge-density of the graphs under consideration and the types of queries which will be allowed.

In the *dense-graph model* ([GGR98], the graph is viewed as a function $f : [N] \times [N] \to \{0, 1\}$. In other words, the tester is given access to the graph via its adjacency matrix. Note that the domain has size N^2 and so a graph is ϵ-far from a given property if at least ϵN^2 edges must be added or removed to "correct" the graph. This model is most appropriate for situations in which graphs typically have degree $\Omega(N)$.

The *bounded-degree model* ([GR97]), on the other hand, is used for constant-degree graphs. In this model, the graph is represented via a series of incidence lists. Formally, the tester has access to a function $f : [N] \times [d] \to [N] \cup \{\perp\}$, where $f(x,i) = y$ means that y is the i-th neighbor of x and $f(x,i) = \perp$ means that x has less than i neighbors. Since the domain has size dN, a graph is ϵ-far from a given property if at least ϵdN edges must be added or removed to "correct" the graph. When analyzing the efficiency of a tester in the bounded-degree model, d is generally viewed as a (small) constant.

More recently, [PR02] and [KKR04] have proposed a *general-graph model* in which the tester is given query access to both the adjacency-matrix and the incidence-list representations of the input graph. Distance, in this model, is measured with respect to $|E|$, which means that the absolute number of allowable "incorrect" edges cannot be *a priori* bounded.

In this thesis, we focus exclusively on the dense-graph model.

1.2 Adaptivity, Non-adaptive, and Canonical Algorithms

Query algorithms can be broadly classified into two types according to how the queries are determined. In an *adaptive algorithm*, the results of previous queries may be used when determining which query to make next. A *non-adaptive algorithm*, on the other hand, determines all of its queries in advance.

In general, adaptive algorithms require far fewer queries than non-adaptive algorithms. For example, approximating the threshold value of a step function to within ϵ requires only $O(\lg \epsilon^{-1})$ adaptive queries (by binary search) but requires $\Omega(\epsilon^{-1})$ non-adaptive queries (since there must be a query within every interval of diameter 2ϵ). Indeed, in the bounded-degree model, any graph property which does not depend solely on the degree distribution requires $\Omega(\sqrt{n}/d)$ queries ([RS06]).

In the dense graph model, however, the maximum gap between the adaptive and non-adaptive query complexities are separated by at most a quadratic factor.

Theorem 2 (canonical-testers)**.** *Let Π be a property which is testable in the dense-graph model using at most $q = q(N, \epsilon)$ (adaptive) queries. Then there exists a non-adaptive tester for Π with query complexity $O(q^2)$. Furthermore, this non-adaptive tester operates by querying the entire subgraph induced by $2q$ random vertices.*

The theorem follows immediately from the observation that any algorithm that makes q queries touches at most $2q$ vertices. By applying a random permutation to G before running the adaptive algorithm, the set of touched vertices is uniformly random. It therefore suffices to query a random induced subgraph and then simulate the adaptive algorithm locally. An algorithm of this form is known as a *canonical algorithm.*

Let q denote the adaptive query-complexity, Q denote the non-adaptive query complexity, and $\widetilde{Q}$ denote the canonical query complexity. By definition, $q \leq Q \leq \widetilde{Q}$. Theorem 2 shows that $\widetilde{Q} = O(q^2)$. A natural question, then, is to determine the exact relationship between these parameters.

It is easy to see that there exist properties with $\widetilde{Q} = \Omega(Q^2)$. Specifically, let Π consist solely of the empty graph. Querying $\Omega(\epsilon^{-2})$ random edges clearly suffices to differentiate the empty graph from one with at least ϵN^2 edges. However, any canonical algorithm must touch $\Omega(\epsilon^{-1})$ vertices, and therefore has query complexity $\Omega(\epsilon^{-2})$. Indeed, [GR09] showed that any non-trivial property has canonical query complexity $\Omega(\epsilon^{-2})$.

Similarly, we can consider the relative power of adaptive versus non-adaptive (but not necessarily canonical) property testers. It is widely believed that the canonical transformation remains optimal, in the worst case, even for this more modest goal. In other words, it is believed that there exist properties such that $Q = \Omega(q^2)$. However, proving such a separation between the adaptive and non-adaptive complexities has proven elusive – no unconditional gap was known until Goldreich and Ron demonstrated a property where $Q = \widetilde{\Omega}(q^{3/2})$ in [GR09].

In light of this difficulty, researchers have considered two modifications of the problem. The first approach, used by [GR07], considers *proximity-dependent* properties, properties which depend (naturally) on the tolerance parameter ϵ. In particular, they considered the combined property of being bipartite and having degree at most $O(\epsilon N)$, and achieved a gap of $Q = \widetilde{\Theta}(q^{4/3})$.

The second approach, used by [GR09], is to consider *promise problems.* In this context, the authors showed that there exist properties such that $Q = \Omega(q^{2-\delta})$ for all $\delta > 0$, exhibiting the first nearly-quadratic gap between adaptive and non-adaptive queries in the dense-graph model. More specifically, the authors demonstrated a hierarchy of gaps of the form $Q = \Theta(q^{2-2/t})$ for each integer $t \geq 2$. Unfortunately, the promise they use, while natural, is quite strong, and it is currently unclear how to remove the promise.

In this thesis, we achieve a nearly-quadratic gap without using a promise, at the expense of making the properties proximity-dependent. This proves that for all $\delta > 0$ and arbitrarily small ϵ, there exists a property Π (which depends on ϵ), such that ϵ-testing Π requires $Q > q^{2-\delta}$ non-adaptive queries, where q is the adaptive query complexity. As in [GR09], we also strengthen the result and establish a hierarchy of relationships between the adaptive and non-adaptive query complexities.

Theorem 3 (Main Theorem)**.** *For all $t \geq 8$ and arbitrarily small ϵ, there exists a graph property Π (which depends on ϵ) such that ϵ-testing Π has non-adaptive query complexity $Q = \widetilde{\Theta}(q^{2-2/t})$, where $q = \widetilde{\Theta}(\epsilon^{-1})$ is the adaptive query complexity.*

Theorem 3 and [GR09] both provide strong evidence that the canonical transformation is optimal in the general case. Although each result technically leaves open the possibility of a better transformation between adaptive and non-adaptive testers, each does so in a different, and very restricted, way. As a result, any such transformation would have to be very unnatural and would have to depend sensitively on the internal structure of the adaptive tester.

1.3 Graph Blow-Ups and Blow-Up Collections

Informally, a *graph blow-up* consists of replacing each vertex of a graph with a cluster of vertices and replacing each edge with a complete bipartite graph (a rigorous definition is given in section 2.1). This operation has been used frequently in studying the dense-graph model, for constructing both upper- and lower-bounds (see, for example, [Alo01], [AS06], and [GKNR09]).

The complexity of testing whether a graph is a blow-up of a fixed graph H was essentially resolved in [Avi09] (see also [AG11]), where it is shown that, for any H, the adaptive query complexity is $O(\epsilon^{-1})$ and the non-adaptive query complexity is $\widetilde{O}(\epsilon^{-1})$.

A graph is a *blow-up collection* if it can be partitioned into disjoint subgraphs, each of which is a blow-up of H. This notion was implicitly introduced in [GR09], which showed the following lower bound.

Lemma 4 ([GR09], Lemma 5.6)**.** *Let H be a simple t-cycle, with $t \geq 4$. Testing whether G is a blow-up collection of H requires $\Omega(\epsilon^{-(2-2/t)})$ non-adaptive queries, even given the promise that G has maximum degree $2t\epsilon N$.*

The authors also showed a lower bound of $\Omega(\epsilon^{-4/3})$ when H is a single loop.

To prove a nearly-quadratic gap between adaptive and non-adaptive query complexities, it therefore suffices to show an $\widetilde{O}(\epsilon^{-1})$ upper-bound on the adaptive query complexity for testing whether a graph is a blow-up collection, for arbitrary H. In [GR09], the authors showed such an upper bound for the cases where H is a single loop and H is a single edge. Unfortunately, they were unable to show such an upper bound for larger H.

In this thesis, we prove tight upper-bounds for both the adaptive and non-adaptive cases under the promise that G has maximum degree $O(\epsilon N)$. Specifically, we show that a tester can determine whether G is a blow-up collection of any given H, given that promise on the degrees, using only $O(\epsilon^{-1})$ adaptive queries or $O(\epsilon^{-2+1/(\Delta+2)} + \epsilon^{-2+2/W})$ non-adaptive queries, where Δ and W are parameters depending only on H. When H is a simple t-cycle, $\Delta = 2$ and $W = t$, and the non-adaptive upper bound reduces to $O(\epsilon^{-2+2/t})$, matching the lower bound in Lemma 4.

Indeed, our result holds even if G is only $O(\epsilon)$-close to satisfying the promise. Since [GR07] gives an efficient non-adaptive tester for the property of having maximum degree $O(\epsilon N)$, we obtain the following two theorems.

Theorem 5 (Adaptive Tester). *For all graphs H and constants $c > 1$, there exists an adaptive property tester (with two-sided error) for the (proximity-dependent) combined property of having maximum degree $c\epsilon N$ and being a blow-up collection of H. The tester has query complexity $O(\epsilon^{-1}\lg^3 \epsilon^{-1})$.*

Since any tester must make $\Omega(\epsilon^{-1})$ queries, Theorem 5 is optimal up to a polylogarithmic factor.

Theorem 5, combined with Lemma 4, suffices to show a gap of size $Q = \Omega(q^{2-\delta})$ for all $\delta > 0$. In the following theorem, we strengthen this result by proving a tight upper-bound on the non-adaptive query complexity. This shows that there is an infinite hierarchy of achievable relationships between the adaptive and non-adaptive query complexities of proximity-dependent properties.

Theorem 6 (Non-Adaptive Tester). *For all graphs H and constants $c > 1$, there exists a non-adaptive property tester (with two-sided error) for the (proximity-dependent) combined property of having maximum degree $c\epsilon N$ and being a blow-up collection of H. The tester has query complexity $O(\epsilon^{-2+1/(\Delta+2)} + \epsilon^{-2+2/W})$, where $\Delta = \deg(H)$ is the maximum degree of H and $W < |H|^2$ is a bound on the size of a witness against H (see Definition 12).*

The proofs of Theorems 5 and 6 are given in Chapter 2.

As mentioned previously, when H is a simple t-cycle, $\Delta = 2$ and $W = t$. Therefore, combining Theorems 5 and 6 with Lemma 4 yields Theorem 3.

Treating c and $|H|$ as constants, we note that the algorithms in both theorems have running time polynomial in the query complexity. In the adaptive case, the query complexity can also be made polynomial in c and $|H|$, although we do not do so here.

Chapter 2

A Hierarchy of Tradeoffs Between Adaptive and Non-adaptive Query Complexities

In this chapter, we prove theorems 5 and 6, thereby establishing theorem 3.

2.1 Notation and Basics

All graphs are assumed to be undirected. Following standard graph-theoretic notation, we let $\Gamma(v) = \{u : (u,v) \in E\}$ denote the neighbors of v. Given $S \subset V$, we let $G|_S$ denote the subgraph induced by S and $\Gamma_S(v) = \Gamma(v) \cap S$ denote the neighbors of v in S. Given $S, T \subset V$, we let $E(S,T) = \{(u,v) \in E : u \in S, v \in T\}$ denote the set of edges between S and T and $S \bigtriangleup T$ denote the symmetric difference of S and T.

Definition 7 (Graph Blow-Up)**.** *A graph $G = ([N], E)$ is a* blow-up *of the graph $H = ([h], F)$ if there exists a partition of $[N]$ into $V_1, \ldots, V_h$ such that for every $i, j \in [h]$ and $(u,v) \in V_i \times V_j$, $(u,v) \in E$ if and only if $(i,j) \in F$. We denote the set of blow-ups of H by $\mathcal{BU}(H)$.*

Note that no requirement is made as to the relative sizes of the V_j. In particular, we allow the case where $|V_j| = 0$. Also note that H is allowed to contain self-loops, in which case G is obtained from H by replacing the vertices with self-loops with cliques instead of independent sets.

Definition 8 (Blow-Up Collection)**.** *A graph $G = ([N], E)$ is a* blow-up collection *of the graph H if there exists a partition of $[N]$ into $V^1, \ldots, V^k$, for some k, such that $V^i \in \mathcal{BU}(H)$ for all i and $E(V^i, V^j) = \emptyset$ for all $i \neq j$. We denote the set of blow-up collections of H by $\mathcal{BUC}(H)$.*

When H is a single loop, [GR09] refers to $\mathcal{BUC}(H)$ as *clique collection*. Similarly, when H is a single edge, they refer to $\mathcal{BUC}(H)$ as BICLIQUE COLLECTION.

Throughout this thesis, H will be an arbitrary fixed graph and $c > 1$ will be an arbitrary fixed constant. We let $h = |H|$ denote the number of vertices in H and $\Delta = \deg(H)$ denote the maximum degree of H.

Definition 9 (Low-Degree Graph)**.** *Given $c > 1$ and $\epsilon > 0$, let $\mathcal{LD}_{c\epsilon} = \{G : \deg(G) \leq c\epsilon N\}$ denote the set of graphs with maximum degree $c\epsilon N$.*

To prove Theorems 5 and 6, we will repeatedly use the concept of a vertex being (k, α)-*partitionable*. Informally, a vertex is (k, α)-partitionable if almost all of its neighbors can be partitioned into k groups, such that all of the vertices within a part have essentially the same neighbors in G. The formal definition is given in Definition 11, after we introduce notation for such partitions.

Definition 10. *Given $v \in V$ and $\alpha > 0$, let $C_{v,\alpha}(u) = \{w \in \Gamma(v) : |\Gamma(w) \vartriangle \Gamma(u)| < \alpha\epsilon N\}$.*

In other words, $C_{v,\alpha}(u)$ consists of all vertices in $\Gamma(v)$ which have "essentially the same" neighbors as u. When v and α are clear from context, we will omit the subscripts and write $C(u)$ for $C_{v,\alpha}(u)$.

Definition 11 ((k, α)-Partitionable)**.** *A vertex v is (k, α)*-partitionable *if there exist representatives $u_1, \ldots, u_k \in \Gamma(v)$ such that $\left|\bigcup_{i=1}^{k} C_{v,\alpha}(u_i)\right| \geq |\Gamma(v)| - \alpha\epsilon N$.*

Note that if $G \in \mathcal{BUC}(H)$, then every vertex is $(\deg(H), 0)$-partitionable.

Finally, we need the idea of a minimal witness against $\mathcal{BUC}(H)$. Informally, a minimal witness is a set of vertices such that the induced subgraph proves that G is not a valid blow-up collection, while any subset does not suffice.

Definition 12 (Minimal Witness)**.** *A set $S \subseteq V$ is a* minimal witness *against $\mathcal{BUC}(H)$ if $G|_S \notin \mathcal{BUC}(H)$, but for any $S' \subsetneq S$, $G|_{S'} \in \mathcal{BUC}(H)$.*

Note that, for any $S \subseteq G$, $G|_S \notin \mathcal{BUC}(H)$ implies that $G \notin \mathcal{BUC}(H)$. Furthermore, any minimal witness against $\mathcal{BUC}(H)$ must be connected, since we could otherwise replace the witness with one of its connected components.

Given H, we let $W = W(H)$ denote the maximum size of a minimal witness. It is easy to see that $W < (h + 1) + \binom{h+1}{2} < h^2$. When H is a simple t-cycle, $t \geq 4$, $W = |H|$.

Handling Technicalities Throughout this thesis, we allow some small informalities for ease of exposition. For example, we often state that an event occurs "with high probability," without specifying the exact failure probability. This is taken to mean that the probability of failure is small enough that taking a union-bound over all (constantly-many) such cases yields a total failure probability of at most, say, 0.01. Similarly, when selecting a random set of vertices, we treat the sampling as being performed with replacement. Since these sets will always be of size $O(1)$ and we will only perform such sampling $O(1)$ times (as a function of N), the odds of a collision are negligible.

2.2 Adaptively Testing $\mathcal{BUC}(H)$ Given a Promise

We begin by showing how to adaptively test whether a graph G is in $\mathcal{BUC}(H)$, given the promise that G is $O(\epsilon)$-close to $\mathcal{LD}_{c\epsilon}$. We view G and ϵ as inputs to the algorithm, and H and c as fixed parameters.

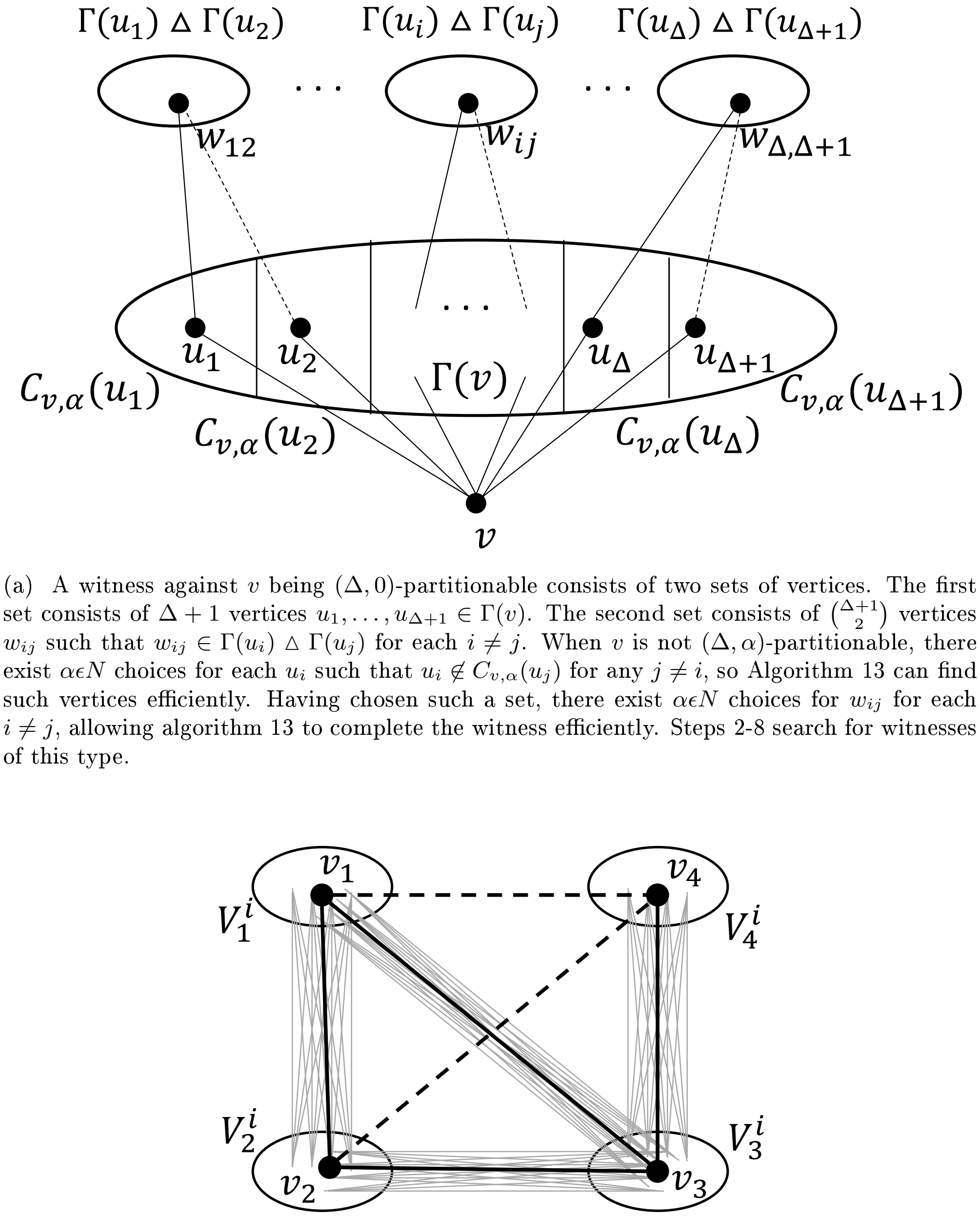
The algorithm consists of two stages. The first stage (steps 2-8 of Algorithm 13) tests whether most vertices are $(\Delta, 0)$-partitionable, where $\Delta = \deg(H)$. To do so, it repeatedly selects a vertex and then attempts to find $\Delta + 1$ neighbors of that vertex which have mutually distinct neighborhoods (see Figure 2.1(a)). If it finds such a set of vertices, they act as a witness against being in $\mathcal{BUC}(H)$ and the algorithm can reject with certainty in step 7.

If the algorithm fails to find such a witness, then most vertices must be (Δ, α)-partitionable, where α is a small constant. This implies that the graph must be close to being a blow-up of some base graph. In other words, almost all of the vertices can be clustered such that each pair of clusters is either almost disjoint or almost forms a complete bipartite graph. The second stage of the algorithm (steps 9-16) checks if this high-level structure is consistent with $\mathcal{BUC}(H)$ (see Figure 2.1(b)). It does so by performing a random search in G for $W = W(H)$ steps, where each step selects a random neighbor of the previously selected vertices. If the resulting W vertices form a witness against $\mathcal{BUC}(H)$, the algorithm rejects in step 15.

Algorithm 13. AdaptiveBlowUpCollectionTest$_{H,c}(G; \epsilon)$

```
1: Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta\,|H|^2)^{-1}$.

   \\ Test whether most vertices are $(\Delta, 0)$-partitionable.
2: for $O(c)$ iterations do
3:     Select a random vertex $v$.
```



(a) A witness against v being $(\Delta, 0)$-partitionable consists of two sets of vertices. The first set consists of $\Delta + 1$ vertices $u_1, \ldots, u_{\Delta+1} \in \Gamma(v)$. The second set consists of $\binom{\Delta+1}{2}$ vertices w_{ij} such that $w_{ij} \in \Gamma(u_i) \triangle \Gamma(u_j)$ for each $i \neq j$. When v is not (Δ, α)-partitionable, there exist $\alpha\epsilon N$ choices for each u_i such that $u_i \notin C_{v,\alpha}(u_j)$ for any $j \neq i$, so Algorithm 13 can find such vertices efficiently. Having chosen such a set, there exist $\alpha\epsilon N$ choices for w_{ij} for each $i \neq j$, allowing algorithm 13 to complete the witness efficiently. Steps 2-8 search for witnesses of this type.

(b) If G has not been rejected by the first half of Algorithm 13, then V can be partitioned into clusters V_j^i, such that each pair of clusters either nearly forms a complete bipartite graph or is nearly disjoint. A minimal witness against having the correct high-level structure consists of a representative vertex from each set such that the induced subgraph is inconsistent with $\mathcal{BUC}(H)$. For example, when $H = K_4$ is the complete graph on 4 vertices, the four vertices shown here form a minimal witness. Steps 9-16 search for witnesses of this type.

Figure 2.1: The two types of witnesses used by Algorithm 13 to determine whether G is a member of $\mathcal{BUC}(H)$. In each diagram, a line indicates that the algorithm queried that edge, with a solid line indicating the presence of an edge and a dashed line indicating the absence of an edge.

```
4:     Select a random set $S$ of $O(\Delta\alpha^{-1}\epsilon^{-1})$ vertices, and query $\{v\} \times S$.
5:     Let $\overline{S}$ be a random subset of $\Gamma_S(v)$ of size (at most) $c\Delta\alpha^{-1}$.
6:     Select a random set $T$ of $O(\Delta^2\alpha^{-1}\epsilon^{-1})$ vertices, and query $\overline{S} \times T$.
7:     If the current view of $G$ is inconsistent with $\mathcal{BUC}(H)$, Reject. Specifically, reject if there
       exist $u_1, \ldots, u_{\Delta+1} \in \overline{S}$ such that $\Gamma_T(u_i) \neq \Gamma_T(u_j)$ for all $i, j$.
8:  end for

    \\ If $G$ has not yet been rejected, most vertices must be $(\Delta, \alpha)$-partitionable.
    \\ Test that $G$ has a high-level structure consistent with $H$.
9:  for $O(Wc)^{O(W)}$ iterations do
10:    Select $v_1$ at random, and let $U = \{v_1\}$.
11:    for $j = 2$ to $W$ do
12:       Select a random set $T_j$ of $O(\Delta\epsilon^{-1})$ vertices, and query $U \times T_j$.
13:       If $\Gamma_{T_j}(U) = \emptyset$, break. Otherwise, randomly select $v_j \in \Gamma_{T_j}(U)$ and let $U = U \cup \{v_j\}$.
14:    end for
15:    If $G|_U \notin \mathcal{BUC}(H)$, Reject.
16: end for
17: If the algorithm hasn't yet rejected, Accept.
```

Algorithm 13 has query complexity $O(\epsilon^{-1})$, as desired. Furthermore, it only rejects if it finds a witness against $\mathcal{BUC}(H)$, so the algorithm accepts valid blow-up collections with probability 1. The following lemma asserts that it rejects graphs which are far from $\mathcal{BUC}(H)$ with high probability.

Lemma 14. *Let G be $\frac{\epsilon}{8}$-close to $\mathcal{LD}_{c\epsilon}$ and ϵ-far from $\mathcal{BUC}(H)$. Then Algorithm 13 rejects with probability at least $2/3$.*

The proof is the content of section 2.2.1.

2.2.1 Proof of Lemma 14

We first note that if G contains many vertices which are not (Δ, α)-partitionable, then G is rejected with high probability.

Lemma 15. *Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta\,|H|^2)^{-1}$. Suppose that G is $\frac{\epsilon}{8}$-close to $\mathcal{LD}_{c\epsilon}$ and contains at least $\frac{1}{4c}N$ vertices which are not (Δ, α)-partitionable. Then Algorithm 13 rejects in step 7 with probability at least $2/3$.*

Proof. Consider the $\frac{1}{4c}N$ vertices which are not (Δ, α)-partitionable. By assumption, at most $\frac{1}{8c}N$ of them have degree greater than $3c\epsilon N$. For otherwise, $2c\epsilon N$ edges would have to be deleted from each such vertex to make G low-degree, meaning that G would have distance at least $\frac{1}{2} \cdot 2c\epsilon N \cdot \frac{1}{8c}N = \frac{\epsilon}{8}N^2$ from $\mathcal{LD}_{c\epsilon}$.

Therefore, with high probability, some iteration chooses a vertex v in step 3 that is not (Δ, α)-partitionable and has degree at most $3c\epsilon N$. Consider that iteration. Note that since v is not (Δ, α)-partitionable, $|\Gamma(v)| > \alpha\epsilon N$.

We view S in step 4 of the algorithm as $S_1 \cup \ldots \cup S_{\Delta+1}$, with $|S_i| = \alpha^{-1}\epsilon^{-1}$. With high probability, $\Gamma_{S_1}(v) \neq \emptyset$, so let u_1 be an arbitrary vertex in $\Gamma_{S_1}(v)$. Since v is not (Δ, α)-partitionable, $\Gamma(v)$ contains at least $\alpha\epsilon N$ vertices u' such that $|\Gamma(u_1) \vartriangle \Gamma(u')| > \alpha\epsilon N$, and so with high probability $\Gamma_{S_2}(v)$ contains such a vertex. Let u_2 be that vertex. We continue in this manner, selecting vertices $u_3, \ldots, u_{\Delta+1}$ such that $|\Gamma(u_i) \vartriangle \Gamma(u_j)| > \alpha\epsilon N$ for all $i \neq j$.

If $|\Gamma_S(v)| < c\Delta\alpha^{-1}$, then $\overline{S} = \Gamma_S(v)$. Otherwise, the probability that each vertex in $\overline{S}$ is a valid choice for u_i, given $u_1, \ldots, u_{i-1}$, is at least $\frac{\alpha}{3c}$, so with high probability $\overline{S}$ contains the desired $\Delta+1$ vertices.

Viewing T as $\bigcup_{i,j=1}^{\Delta+1} T_{ij}$, we see that, with high probability, T contains a vertex in $\Gamma(u_i) \vartriangle \Gamma(u_j)$ for all $i \neq j$. The resulting view of G is therefore inconsistent with $\mathcal{BUC}(H)$ and the algorithm rejects, as desired. □

We now show that if G has at most $\frac{1}{4c}N$ vertices which are not (Δ, α)-partitionable, then G can be partitioned into components such that, for each pair of components, either the edges between them almost form a complete bipartite graph or the components are almost disjoint.

Lemma 16. *Let* $\Delta = \deg(H)$ *and* $\alpha = (16\Delta |H|^2)^{-1}$. *Suppose that* G *is* $\frac{\epsilon}{8}$*-close to* $\mathcal{LD}_{c\epsilon}$ *and contains at most* $\frac{1}{4c}N$ *vertices which are not* (Δ, α)*-partitionable. Then* G *is* $\frac{5\epsilon}{8}$*-close to a graph* $\widetilde{G} = (V, \widetilde{E})$ *for which the following holds:* V *can be partitioned into* $\bigcup_{i,j} V_j^i \cup L$ *such that*

1. $\widetilde{\Gamma}(v) = \emptyset$ *for all* $v \in L$,
2. $E(V^i, V^{i'}) = \emptyset$ *for all* $i \neq i'$,
3. *For all* i, j, $\widetilde{\Gamma}(u) = \widetilde{\Gamma}(v)$ *for all* $u, v \in V_j^i$, *and*
4. $\left|V_j^i\right| > \frac{\epsilon}{16\Delta}N$ *for all* i, j,

where $V^i = \bigcup_j V_j^i$ *and* $\widetilde{\Gamma}(u)$ *is the neighborhood of* u *in* $\widetilde{G}$. *Furthermore,*
$\left|\Gamma_{V^i}(u) \vartriangle \widetilde{\Gamma}_{V^i}(u)\right| < \alpha\epsilon N$ *for all* $u \in V^i$.

Proof. First, delete all edges adjacent to any vertex which is not (Δ, α)-partitionable and add those vertices to L. The total cost of doing so is at most $\frac{1}{4c}N \cdot c\epsilon N + \frac{\epsilon}{8}N^2 = \frac{3\epsilon}{8}N^2$, since G is $\frac{\epsilon}{8}$ close to

$\mathcal{LD}_{c\epsilon}$.

The construction now proceeds in stages, with each stage constructing a $V^i = \bigcup_j V_j^i$ such that $\widetilde{E}(V^i, V \backslash V^i) = \emptyset$.

To construct $V^1 = \bigcup_j V_j^1$, choose an arbitrary vertex v with $|\Gamma(v)| > \frac{\epsilon}{8}N$ and (Δ, α)-partition $\Gamma(v)$ into $\bigcup_{j=1}^{\Delta} C(u_j)$. For any j such that $|C(u_j)| > \frac{\epsilon}{16\Delta}N$, add $C(u_j)$ to V^1 as a distinct V_j^i. Note that since $|\Gamma(v)| > \frac{\epsilon}{8}N$, there must be at least one partition of that size.

While there exists a u_j in V^1 such that $\left|\Gamma(u_j) \backslash V^1\right| > \frac{\epsilon}{8}N$, choose such a vertex. For consistency of notation, we relabel u_j to v. Let $\bigcup_{j=1}^{\Delta} C(u_j)$ be a (Δ, α)-partitioning of $\Gamma(v)$. Again, there must be a component such that $\left|C(u_j) \backslash V^1\right| > \frac{\epsilon}{16\Delta}N$. We add those $C(u_j) \backslash V^1$ to V^1 as distinct V_j^is.

Once every u_j in V^1 has $\left|\Gamma(u_j) \backslash V^1\right| < \frac{\epsilon}{8}N$, we are ready to finalize V^1. We begin by separating V^1 from $V \backslash V^1$ by deleting $E(V^1, V \backslash V^1)$, at a cost per vertex of at most $\frac{\epsilon}{8}N + \alpha\epsilon N$. We next ensure that $E(V_j^1, V_{j'}^1)$ is either $V_j^1 \times V_{j'}^1$ or the empty set, for all j, j'. This can be easily shown to require at most $\alpha\epsilon N$ edits per vertex.

Note that V^1 now satisfies conditions 2-5, by construction.

We now repeat this entire process on the remaining vertices, creating V^2, V^3, …, until every vertex in V has either been covered or has degree at most $\frac{\epsilon}{8}N$. Finally, we delete all edges adjacent to the leftover vertices, at a cost of $\frac{\epsilon}{8}N$ per vertex, and add those vertices to L.

The total cost of this procedure is bounded by $(\frac{\epsilon}{8}N + 2\alpha\epsilon N)N < \frac{\epsilon}{4}N^2$, for a final cost of $\frac{5\epsilon}{8}N^2$, as desired. □

Note that the bound on $\left|\Gamma_{V^i}(u) \triangle \widetilde{\Gamma}_{V^i}(u)\right|$ in the lemma implies that $\widetilde{G}$ is $\frac{\epsilon}{8}$-close to having maximum degree $2c\epsilon N$, since G is $\frac{\epsilon}{8}$-close to having maximum degree $c\epsilon N$ and the degree of each vertex increases by at most $\alpha\epsilon N$ when going from G to $\widetilde{G}$. Therefore, conditions 3 and 4 imply that most of the components V_j^i are connected to at most $32c\Delta$ other components and have size at most $2c\epsilon N$.

Intuitively, these conditions allow us to view each cluster V_j^i as a supernode and each bipartite graph as an edge. From this viewpoint, $\widetilde{G}$ becomes a bounded-degree graph, with maximum degree $32c\Delta$. Since each supernode contains $\Omega(\epsilon N)$ vertices (by condition 4 in the lemma), we can simulate neighbor queries by querying $O(\epsilon^{-1})$ random vertices and choosing a random neighbor. Furthermore, if $\widetilde{G}$ is $\Omega(\epsilon)$-far from $\mathcal{BUC}(H)$ when viewed as a dense graph, the corresponding bounded-degree graph is $\Omega(1)$-far from $\mathcal{BUC}(H)$.

To formalize this intuition, we first show that the set of witnesses against $\widetilde{G}$ covers a constant fraction of V.

Lemma 17. *Let $\widetilde{G}$ be as in Lemma 16, and suppose that $\widetilde{G}$ is $\frac{3\epsilon}{8}$-far from $\mathcal{BUC}(H)$. Then there exist at least $(16Wc^2\epsilon)^{-1}$ disjoint sets $\widetilde{W_k} = \bigcup_\ell V_{j_{k,\ell}}^{i_k}$ such that $G|_{\widetilde{W_k}}$ is a witness against $\mathcal{BUC}(H)$.*

Proof. First, delete $\frac{\epsilon}{8}N^2$ edges so that $\widetilde{G}$ is in $\mathcal{LD}_{2c\epsilon}$. Note that the resulting graph is still $\frac{\epsilon}{4}$-far from $\mathcal{BUC}(H)$.

Consider some $\widetilde{W_k}$. Completely disconnecting all vertices contained in $\widetilde{W_k}$ requires at most $W(2c\epsilon N)^2$ deletions, since there are at most W components and, by Lemma 16, each component contains at most $2c\epsilon N$ vertices and each vertex has degree at most $2c\epsilon N$.

Therefore, if there are less than $(16Wc^2\epsilon)^{-1}$ disjoint $\widetilde{W_k}$, the total cost of deleting all witnesses in $\widetilde{G}$ is at most $(16Wc^2\epsilon)^{-1} \cdot W(2c\epsilon N)^2 = \frac{\epsilon}{4}N^2$. This contradicts the assumed distance to $\mathcal{BUC}(H)$, and the lemma follows. □

We are now ready to show that the second half of Algorithm 13 finds a witness against G with high probability.

Informally, by Lemma 17, step 10 of the algorithm selects a vertex $v_1 \in V_{j_1}^i$ corresponding to a witness with high probability. Having chosen v_1, we wish to bound the probability that $v_2, \ldots, v_W$ are chosen so as to form a complete witness. Recalling that any minimal witness is connected, there must exist a $v_2 \in V_{j_2}^i$ which extends the witness in $\widetilde{G}$. By condition 3 of Lemma 16, any vertex in $V_{j_2}^i$ can be used in place of v_2 to extend the witness. However, since $\left|V_{j_2}^i\right| > \frac{\epsilon}{16\Delta}N$ and $\left|\Gamma_{V^i}(u) \vartriangle \widetilde{\Gamma}_{V^i}(u)\right| < \alpha\epsilon N < \frac{1}{2W}\left|V_{j_2}^i\right|$, at least half of $V_{j_2}^i$ is a valid choice to extend the witness in G. Furthermore, most of these must have degree $O(c\epsilon N)$.

Iterating this procedure, we see that there are always at least $\Omega(\frac{\epsilon}{\Delta}N)$ choices for v_j which extend the witness in G. Furthermore, since each previous $v_{j'}$ was chosen to have degree $O(c\epsilon N)$, there are at most $O(Wc\epsilon N)$ potential choices for v_j. Therefore, with probability $\Omega((W^2c)^{-1})$, Algorithm 13 chooses a good v_j at each step and finds a complete witness in G.

The formal proof is as follows.

Proof of Lemma 14. If G is rejected with probability at least $2/3$ by the first part of Algorithm 13, then we are done. So suppose otherwise. Then by Lemma 15, there exists $\widetilde{G}$ satisfying the conclusion of Lemma 16. Since G is ϵ-far from $\mathcal{BUC}(H)$, $\widetilde{G}$ is at least $\frac{3\epsilon}{8}$-far from $\mathcal{BUC}(H)$.

By Lemma 17, there exist $(16Wc^2\epsilon)^{-1}$ sets $\widetilde{W_k} = \bigcup_\ell V^{i_k}_{j_{k,\ell}}$ corresponding to witnesses. Call a $\widetilde{W_k}$ *high-degree* if, for some $V^{i_k}_{j_{k,\ell}} \subset \widetilde{W_k}$ contains $\frac{\epsilon}{64\Delta}N$ vertices of degree greater than $65\Delta Wc^2\epsilon N$. We note that at most half of the $\widetilde{W_k}$ are high-degree. Otherwise, G would have distance at least $\frac{1}{2} \cdot (16Wc^2\epsilon)^{-1} \cdot 64\Delta Wc^2\epsilon N \cdot \frac{\epsilon}{64\Delta}N = \frac{\epsilon}{32}N^2$ to $\mathcal{LD}_{c\epsilon}$, contrary to assumption.

We therefore restrict our attention to the $(32Wc^2\epsilon)^{-1}$ witnesses which are not high-degree.

Let $V^i_1, \ldots, V^i_\ell$, $\ell < W$, be components corresponding to a partial witness in $\widetilde{G}$. In other words, $(V^i_1, \ldots, V^i_\ell) = (W^{i^*}_{j_1}, \ldots, W^{i^*}_{j_\ell})$, for some i^* and $j_1, \ldots, j_\ell$. Let $v_1, \ldots, v_\ell$ be arbitrary vertices in $V^i_1, \ldots, V^i_\ell$, respectively, such that $|\Gamma(v_j)| < 65\Delta Wc^2\epsilon N$ for each $j \leq \ell$. Let $V^i_{\ell+1}$ be a component adjacent to $\bigcup_{j=1}^{\ell} V^i_j$ which extends the partial witness in $\widetilde{G}$. Such a component must exist, since witnesses are connected.

Since $\left|V^i_{\ell+1}\right| > \frac{\epsilon}{16\Delta}N$, there are at least $\frac{\epsilon}{16\Delta}N$ vertices which extend the witness (in $\widetilde{G}$). Recall that if V^i_j is adjacent to $V^i_{j'}$ in $\widetilde{G}$, then every vertex in V^i_j must be adjacent in G to all but at most $\alpha\epsilon N$ vertices in $V^i_{j'}$ (and similarly if V^i_j is not adjacent to $V^i_{j'}$). Therefore, since $W < \frac{1}{2}\left|H\right|^2$ and $\alpha = (16\Delta\left|H\right|^2)^{-1}$, there must be at least $\frac{\epsilon}{16\Delta}N - W\alpha\epsilon N > \frac{\epsilon}{32\Delta}N$ choices for $v_{\ell+1} \in V^i_{\ell+1}$ such that $E(v_j, v_{\ell+1}) = \widetilde{E}(v_j, v_{\ell+1})$ for all $j \leq \ell$, which means that $v_{\ell+1}$ extends the witness in G as well. Of these, at least $\frac{\epsilon}{64\Delta}N$ must also have degree at most $65\Delta Wc^2\epsilon N$.

Since $|T_i| = O(\Delta^{-1}\epsilon^{-1})$, with high probability $\Gamma_{T_i}(U)$ contains such a vertex in step 13. Since each of the ℓ vertices selected so far is adjacent to at most $65c\epsilon N$ vertices, there are at most $\ell 65c\epsilon N$ candidates for $v_{\ell+1}$, and so the probability that the selected $v_{\ell+1}$ extends the witness in G is at least $\frac{\epsilon/32\Delta}{\ell 65c\epsilon} = \Omega((W^2c)^{-1})$.

Therefore, with high probability, the algorithm finds a complete witness in some iteration of steps 10-14 and rejects G. □

2.3 Non-adaptively Testing $\mathcal{BUC}(H)$ Given a Promise

We now show how to non-adaptively test whether a graph is in $\mathcal{BUC}(H)$, given the promise that G is $O(\epsilon)$-close to $\mathcal{LD}_{c\epsilon}$.

As in the adaptive case, the first stage of the algorithm verifies that most vertices are (Δ, α)-

partitionable. Assuming the graph passes the first stage, the second stage checks that the high-level structure of G is consistent with $\mathcal{BUC}(H)$.

Since we can no longer adaptively restrict our queries to neighbors of a given vertex, we instead rely on the birthday paradox to achieve sub-quadratic query complexity. Recall that the birthday paradox says that, given a discrete domain D, $O(|D|^{1-1/k})$ uniformly-chosen samples suffice to obtain a k-wise collision with high probability.

We give a slightly generalized version here that will be useful for the following proofs.

Lemma 18 (Birthday Paradox)**.** *Let $\mathcal{D}$ be a finite domain, and let $\{\mu_i\}$ be a set of probability distributions over $\mathcal{D}$ with $\mu_i(d) = \Omega(1/|\mathcal{D}|)$ for all i and $d \in \mathcal{D}$. Suppose that the i-th sample is drawn according to μ_i. Then $O(|\mathcal{D}|^{1-1/k})$ samples suffice to obtain a k-wise collision with high probability.*

Proof. By assumption, there exists a fixed $0 < c < 1$ such that $\mu_i(d) > \frac{c}{|\mathcal{D}|}$ for all i and $d \in \mathcal{D}$. Let $\alpha = \frac{c}{|\mathcal{D}|}$.

Considering the following process. At step i, we select a sample according to μ_i. For each $d \in \mathcal{D}$, we keep the sample with probability $\frac{\alpha}{\mu_i(d)}$ and discard it otherwise. Note that this process is equivalent to sampling from the distribution $\widetilde{\mu}$ over $\mathcal{D} \cup \{\perp\}$ which selects $d \in D$ with probability α and $\perp$ with probability $1 - c$ and discarding any $\perp$s.

With high probability, $2c^{-1}M$ samples from $\widetilde{\mu}$ suffice to obtain M non-discarded samples. Since the non-discarded samples are uniformly distributed, it follows from the standard birthday paradox that $2c^{-1} \cdot O(|D|^{1-1/k}) = O(|D|^{1-1/k})$ samples from $\widetilde{\mu}$ suffice to obtain a k-wise collision with high probability.

Since discarding samples can only increase the number of samples required to achieve a k-wise collision, the lemma follows. □

Very informally, in our setting a collision will correspond to choosing multiple vertices from a single witness in such a way that a W-wise collision corresponds to a complete witness. Assuming that the set of disjoint witnesses has size $O(\epsilon^{-1})$ and that we can sample from that set with only constant overhead, the birthday paradox implies that $O(\epsilon^{-1+1/W})$ random vertices suffice to find all W vertices corresponding to a complete witness. Querying the entire induced subgraph then yields the desired result. The primary challenge, therefore, is to show that when G is ϵ-far from $\mathcal{BUC}(H)$, the algorithm can efficiently sample from the space of witnesses.

The structure of the non-adaptive algorithm parallels the structure of the adaptive algorithm, especially with regards to proving correctness. Steps 2-4 of Algorithm 19 correspond to steps 2-8 of Algorithm 13 and check that most vertices are $(\Delta, 0)$-partitionable, where $\Delta = \deg(H)$. Steps 5-6 correspond to steps 9-16 of Algorithm 13 and check that the high-level structure of G is consistent with $\mathcal{BUC}(H)$.

Algorithm 19. $\textsc{NonAdaptiveBlowUpCollectionTest}_{H,c}(G;\epsilon)$

```
1: Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta\,|H|^2)^{-1}$.
 \\ Check that most vertices are $(\Delta, 0)$-partitionable.
2: Select a set $S$ of $O((\alpha\epsilon)^{-1+1/(\Delta+2)})$ random vertices, and query $S \times S$.
3: Select a set $T$ of $O(\Delta^2\alpha^{-1}\epsilon^{-1})$ random vertices, and query $S \times T$.
4: If the current view of $G$ is inconsistent with $\mathcal{BUC}(H)$, REJECT. Specifically, reject if there exist
   $v \in S$ and $u_1, \ldots, u_{\Delta+1} \in \Gamma_S(v)$ such that $\Gamma_T(u_i) \neq \Gamma_T(u_j)$ for all $i, j$.

 \\ Check that $G$ has a high-level structure consistent with $H$.
5: Select a set $S$ of $O((\Delta c^2\epsilon)^{-1+1/W})$ random vertices and query $S \times S$.
6: If $G|_S \notin \mathcal{BUC}(H)$, REJECT.
7: If the algorithm hasn't yet rejected, ACCEPT.
```

Algorithm 19 has query complexity $O(\epsilon^{-2+1/(\Delta+2)} + \epsilon^{-2+2/W})$, as desired. Since it only rejects if it finds a witness against G, it accepts all graphs in $\mathcal{BUC}(H)$ with probability 1. It remains to show that it rejects graphs which are far from $\mathcal{BUC}(H)$ with high probability.

Lemma 20. *Let $\alpha = (16\Delta\,|H|^{-2})^{-1}$, as in Algorithm 19, and let G be $\frac{\alpha\epsilon}{16c}$-close to $\mathcal{LD}_{c\epsilon}$ and ϵ-far from $\mathcal{BUC}(H)$. Then Algorithm 19 rejects with probability at least $2/3$.*

The proof is the content of section 2.3.1.

2.3.1 Proof of Lemma 20

The proof closely follows the structure of the proof for the adaptive case given in section 2.2.1. Because the arguments used in the non-adaptive case are more sensitive to the degrees of selected vertices than in the adaptive case, we will assume that G is $\frac{\alpha\epsilon}{16}$-close to $\mathcal{LD}_{c\epsilon}$, instead of only being $\frac{\epsilon}{8}$-close as in the adaptive case, where $\alpha < 1$ is a fixed constant.

As in the adaptive case, we begin by noting that if G contains many vertices which are not (Δ, α)-partitionable, then G is rejected with high probability.

Lemma 21. *Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta\,|H|^2)^{-1}$. Suppose that G is $\frac{\alpha\epsilon}{16c}$-close to $\mathcal{LD}_{c\epsilon}$ and contains at least $\frac{1}{4c}N$ vertices which are not (Δ, α)-partitionable. Then Algorithm 19 rejects in step 4 with probability at least $2/3$.*

Proof. For ease of exposition, we begin by proving the claim under the stronger assumption that $G \in \mathcal{LD}_{c\epsilon}$. We then show how to modify the argument to only require that G be $O(\epsilon)$-close to $\mathcal{LD}_{c\epsilon}$.

We view the algorithm as choosing S one vertex at a time until a particular event occurs. We emphasize that this is merely a proof technique, since the algorithm is non-adaptive, and so the actual queries performed will be a superset of the ones described in the proof. Since witnesses are preserved under additional queries, this suffices to prove the lemma.

As the samples are chosen, we will maintain an approximate partitioning of V into $\Theta(\epsilon^{-1})$ sets, such that each vertex belongs to at most Δ sets and each set has size $\Omega(\epsilon N)$, and therefore a constant fraction of V is always covered. Intuitively, selecting a vertex from the k-th set will correspond to selecting the next vertex in the k-th witness, until a complete witness has been found.

Formally, we initially partition the $\frac{1}{4c}N$ vertices which are not (Δ, α)-partitionable into $\bigcup_{k=1}^{(\alpha\epsilon)^{-1}} U_k$, with $|U_k| = \frac{\alpha\epsilon}{4c}N$. For each k, we initially say that U_k is *uninitialized.* The first time that we select $v \in U_k$, we say that U_k becomes *initialized with seed* v, after which we will require that $U_k \subseteq \Gamma(v)$.

We now proceed to sample vertices at random. If the selected vertex u does not belong to any U_k, we discard that sample and try again. If $u \in U_k$ for some k, we update the partitioning as follows. If U_k is uninitialized, then u must not be (Δ, α)-partitionable (by construction), and we set $U_k = \Gamma(u)$. Note that since u is not (Δ, α)-partitionable, $|U_k| > \alpha\epsilon N$ as required. We also update the remaining uninitialized $U_{k'}$ by removing any vertex $v \in U_{k'}$ such that $v \in \Gamma(u)$ or $|\Gamma(u) \triangle \Gamma(v)| < \frac{1}{2}\alpha\epsilon N$. Finally, we rebalance the sizes of the uninitialized $U_{k'}$. Supposing for the moment that $G \in \mathcal{LD}_{c\epsilon}$, we remove at most $|\Gamma(u)| \cdot c\epsilon N / \frac{1}{2}\alpha\epsilon N < 2c^2\alpha^{-1}\epsilon N$ vertices in total from $\bigcup U_k$.

If U_k has been initialized with seed v, we proceed as follows. Let $u_1, \ldots, u_\ell \in \Gamma(v)$, $\ell \le \Delta$, be the (non-seed) vertices which have been selected from U_k so far. We set

$$U_k = \Gamma(v) \backslash \bigcup_{i=1}^{\ell} C_{v,\alpha}(u_i).$$

Note that this ensures that $u_i \in \Gamma(v)$ and $|\Gamma(u_i) \triangle \Gamma(u_j)| > \alpha\epsilon N$ for all i, j. Furthermore, since the seed v is not (Δ, α)-partitionable, we have $|U_k| > \alpha\epsilon N$. All other $U_{k'}$ are unchanged.

We continue this way until either (i) $\Delta + 2$ samples have been chosen from a single U_k or (ii) at least half of the vertices in some initialized U_k belong to at least Δ other initialized $U_{k'}$. We claim that this occurs, with high probability, within $O(\epsilon^{-1+1/(\Delta+2)})$ steps.

To see this, suppose that condition (ii) has not occurred. First note that either $(8c^3\alpha^{-1}\epsilon)^{-1}$ of the U_k have been initialized, or at least $\frac{1}{8c}N$ vertices are contained in the uninitialized U_k. Therefore,

$$\left|\bigcup_k U_k\right| > (8c^3\alpha^{-1}\epsilon)^{-1} \cdot \frac{\alpha\epsilon}{2\Delta}N = \frac{\alpha^2}{8c^3}N = \Omega(N),$$

so $O(1)$ samples from V suffice to sample from $\bigcup_k U_k$ with high probability. By the birthday paradox, it follows that $O((\frac{\Delta}{\alpha}\epsilon)^{-1+1/(\Delta+2)}) = O(\epsilon^{-1+1/(\Delta+2)})$ samples suffice to obtain the desired $(\Delta + 2)$-wise collision, with high probability. In other words, after $O(\epsilon^{-1+1/(\Delta+2)})$ steps, either condition (ii) has occurred or we have selected $v, u_1, \ldots, u_{\Delta+1}$ from some U_k.

Suppose that we selected $v, u_1, \ldots, u_{\Delta+1}$ from some U_k. Then with high probability, T contains a vertex from $\Gamma(u_i) \vartriangle \Gamma(u_j)$, for each i, j, and the algorithm rejects in step 4.

If, instead, condition (ii) occurred, then there exist $\frac{\alpha\epsilon}{2}N$ vertices which are adjacent to $\Delta + 1$ of the seed vertices. With high probability, T contains such a vertex u. Let $v_1, \ldots, v_{\Delta+1}$ be the corresponding seed vertices. Recall that, by construction, $|\Gamma(v_i) \vartriangle \Gamma(v_j)| > \frac{1}{2}\alpha\epsilon N$ for all seed vertices v_i, v_j. Therefore, with high probability, T contains a vertex in $\Gamma(v_i) \vartriangle \Gamma(v_j)$ for each $i \neq j$, and the algorithm rejects in step 4.

Note that in the preceding argument, we only assumed that $G \in \mathcal{LD}_{c\epsilon}$ when bounding the number of vertices discarded after each step from the unitialized $U_{k'}$. Note, however, that each additional discarded vertex implies an additional $\frac{1}{2}\alpha\epsilon N$ distance from $\mathcal{LD}_{c\epsilon}$. Since G was assumed to be $\frac{\alpha\epsilon}{16c}$-close to $\mathcal{LD}_{c\epsilon}$, we therefore delete at most an additional $\frac{1}{16c}N$ vertices from $\bigcup U_k$, in total over all steps. It follows that $|\bigcup_k U_k| = \Omega(N)$, as required, and the rest of the proof goes through unchanged.

□

By Lemma 21, either G is rejected with high probability by step 4 or we can apply Lemma 16 (since

$\frac{\alpha\epsilon}{16c} < \frac{\epsilon}{8}$) to obtain $\widetilde{G}$, as in the adaptive case. By Lemma 17, it follows that there are many disjoint witnesses against $\widetilde{G}$, which allows us to again apply the birthday paradox.

Specifically, let $\mathcal{D} = \{\widetilde{W_k}\}$ be the set of witnesses guaranteed by Lemma 17. As was shown in the adaptive case, for any choice of $v_1, \ldots, v_\ell$ corresponding to some $\widetilde{W_k}$, there exist at least $\frac{\epsilon}{32\Delta}N$ choices for $v_{\ell+1}$ which correctly extend $\widetilde{W_k}$ in G. Therefore, $O(Wc^2\Delta)$ samples suffice to obtain a vertex which extends some witness. Applying the birthday paradox, we see that step 5 finds a complete witness with high probability, in which case the algorithm rejects in step 6.

We now formalize this argument.

Proof of Lemma 20. If G is rejected with probability at least 2/3 by the first part of Algorithm 19, then we are done. So suppose otherwise. Then by Lemma 21, G contains at most $\frac{1}{4c}N$ vertices which are not (Δ, α)-partitionable, and so there exists $\widetilde{G}$ satisfying the conclusion of Lemma 16. So by Lemma 17, there exist at least $(8Wc^2\epsilon)^{-1}$ distinct witnesses $\widetilde{W_k}$.

As in the proof of Lemma 14, let $V_1^i, \ldots, V_\ell^i$, $\ell < W$, be components corresponding to a partial witness in $\widetilde{G}$. In other words, $(V_1^i, \ldots, V_\ell^i) = (W_{j_1}^{i^*}, \ldots, W_{j_\ell}^{i^*})$, for some i^* and $j_1, \ldots, j_\ell$. Let $v_1, \ldots, v_\ell$ be arbitrary vertices in $V_1^i, \ldots, V_\ell^i$, respectively. Let $V_{\ell+1}^i$ be a component adjacent to $\bigcup_{j=1}^{\ell} V_j^i$ which extends the partial witness in $\widetilde{G}$. Such a component must exist, since minimal witnesses are connected.

Since $|V_{\ell+1}^i| > \frac{\epsilon}{16\Delta}N$, there are at least $\frac{\epsilon}{16\Delta}N$ vertices which extend the witness (in $\widetilde{G}$). Also recall that if V_j^i is adjacent to $V_{j'}^i$ in $\widetilde{G}$, then every vertex in V_j^i must be adjacent in G to all but at most $\frac{\epsilon}{16\Delta h^2}$ vertices in $V_{j'}^i$ (and similarly if V_j^i is not adjacent to $V_{j'}^{i'}$). Therefore, since $W < \frac{1}{2}h^2$, there must be at least $\frac{\epsilon}{16\Delta}N - W\frac{\epsilon}{16\Delta h^2}N > \frac{\epsilon}{32\Delta}N$ choices for $v_{\ell+1}$ such that $E(v_j, v_{\ell+1}) = \widetilde{E}(v_j, v_{\ell+1})$ for all $j \leq \ell$, which means that $v_{\ell+1}$ extends the witness in G as well.

We now map step 5 onto the birthday paradox. Given vertices $v_1^k, \ldots, v_\ell^k$ in $V_{j_{k,1}}^{i_k}, \ldots, V_{j_{k,\ell}}^{i_k}$, respectively, let $U_k \subseteq V_{j_{k,\ell+1}}^{i_k}$ be the set of vertices which extend $\widetilde{W_k}$ in G.

We let $\mathcal{D} = \{U_k\}$ and μ_i be the probability distribution induced by selecting vertices uniformly at random. Since $|U_k| > \frac{\epsilon}{32\Delta}N$ for all k and the U_k are disjoint, $O(Wc^2\Delta)$ samples from V suffice to obtain a sample from $\mathcal{D}$. Therefore, by Lemma 18, $O(\epsilon^{-1+1/W})$ samples suffice to obtain a complete witness, which means that S contains a complete witness with high probability and the algorithm rejects in step 6. □

2.4 Removing the Low-Degree Promise

We now show how to test the combined property of being a valid blow-up collection and having low-degree.

In [GR07], the authors give an $\widetilde{O}(\epsilon^{-1})$-query algorithm for testing whether the input has maximum degree $O(\epsilon N)$.

Lemma 22 ([GR07], Theorem 3)**.** *Fix $c > 1$ and $\beta > 0$. There exists a non-adaptive tester with query complexity $\widetilde{O}(\epsilon^{-1})$ and two-sided error which accepts graphs with maximum degree $c\epsilon N$ with probability at least $2/3$ and rejects graphs which are $\beta\epsilon$-far from having maximum degree $c\epsilon N$ with probability at least $2/3$.*

To test whether G is in $\mathcal{BUC}(H) \cap \mathcal{LD}_{c\epsilon}$, we therefore run the tester from Lemma 22, and if it accepts, we then run either the adaptive or non-adaptive tester for $\mathcal{BUC}(H)$.

All that remains is to show that if G is ϵ-far from $\mathcal{BUC}(H) \cap \mathcal{LD}_{c\epsilon}$, then it must be $\Omega(\epsilon)$-far from $\mathcal{BUC}(H)$ or $\Omega(\epsilon)$-far from $\mathcal{LD}_{c\epsilon}$.

Lemma 23. *Suppose that G is $\frac{\epsilon}{18c\Delta^2}$-close to $\mathcal{LD}$ and $\frac{\epsilon}{3}$-close to $\mathcal{BUC}(H)$. Then G is ϵ-close to $\mathcal{LD} \cap \mathcal{BUC}(H)$.*

Proof. Let $\epsilon_1 \leq \frac{\epsilon}{18c\Delta^2}$ be the distance to $\mathcal{LD}_{c\epsilon}$ and $\epsilon_2 \leq \frac{\epsilon}{3}$ be the distance to $\mathcal{BUC}(H)$. Let $V = \bigcup V_j^i$ be an optimal decomposition of G with respect to $\mathcal{BUC}(H)$.

First, we completely disconnect all V_j^i such that $\left|V_j^i\right| < \frac{\epsilon}{3\Delta}N$. Next, we delete all superfluous edges and add all missing edges so that $G \in \mathcal{BUC}(H)$, at total cost at most $\epsilon_2 N^2$. Note that since $\bigcup V_j^i$ is an optimal decomposition, every vertex must have been connected to at least half of its neighbors (otherwise we could reduce the cost by assigning that vertex to its own component). Therefore, at worst this doubled the degree of every vertex, thereby doubling the distance to $\mathcal{LD}$. Note that G is now a valid blow-up collection.

The total cost of the edits so far is bounded by $\epsilon_2 N^2 + \sum \left|V_j^i\right| \Delta \frac{\epsilon}{3\Delta} N = \epsilon_2 N^2 + \frac{\epsilon}{3} N^2$, since each component is adjacent to at most Δ other components.

For each i, j, if $\left|V_j^i\right| > c\epsilon N$, delete $\left|V_j^i\right| - c\epsilon N$ vertices from V_j^i. This clearly preserves membership in $\mathcal{BUC}(H)$. Furthermore, the cost of doing the deletion is exactly equal to the decrease in distance to $\mathcal{LD}_{c\epsilon}$. To see this, note that every neighbor of V_j^i needed to delete at least $\left|V_j^i\right| - c\epsilon N$ into V_j^i to become low-degree, and so without loss of generality, we can assume that they all delete the edges going to the same $\left|V_j^i\right| - c\epsilon N$ vertices.

Finally, while any high-degree vertices remain, we do the following. First, choose $u \in V_j^i$ such that $|\Gamma(u)| > c\epsilon N$ and $v \in \Gamma(u)$, and delete v. Note that the cost of doing so is, at most, $c\epsilon N \cdot \Delta$ and the decrease in distance is at least $\left|V_j^i\right| > \frac{\epsilon}{3\Delta} N$, for a net multiplicative overhead of $3c\Delta^2$.

The total cost of removing the high-degree vertices is therefore bounded by $3c\Delta^2 \cdot 2\epsilon_1 N^2$, which means that the final cost is bounded by $(\epsilon_2 N^2 + \frac{\epsilon}{3} N^2) + (3c\Delta^2 \cdot 2\epsilon_1 N^2) \leq \epsilon N^2$ as desired. □

We now prove Theorems 5 and 6.

Proof of Theorems 5 and 6. We first run the tester from Lemma 22 with $\beta = (18c\Delta(H)^2)^{-1}$. If it accepts, we then run Algorithm 13 (in the adaptive case) or Algorithm 19 (in the non-adaptive case) with tolerance $\frac{\epsilon}{3}$.

If $G \in \mathcal{LD}_{c\epsilon} \cap \mathcal{BUC}(H)$, then the low-degree tester accepts with probability at 2/3 and blow-up collection tester accepts with probability 1. So G is accepted with probability at least 2/3.

Suppose that G is ϵ-far from $\mathcal{LD} \cap \mathcal{BUC}(H)$. By Lemma 23, either G is $\frac{\epsilon}{18c\Delta^2}$-far from $\mathcal{LD}$ or G is $\frac{\epsilon}{3}$-far from $\mathcal{BUC}(H)$. In the first case, the low-degree tester rejects with probability at least 2/3. In the second case, the blow-up collection tester rejects with probability at least 2/3, by Lemma 14 (in the adaptive case) or Lemma 20 (in the non-adaptive case). □ □

Chapter 3

An Alternate Proof that CLIQUECOLLECTION has Adaptive Query Complexity $\widetilde{O}(\epsilon^{-1})$

In [GR09], the authors prove that testing the complexity of $\mathcal{BUC}(H)$ is $\widetilde{O}(\epsilon^{-1})$, without any promise on the degree bound, for the special cases where H is a self-loop and or H is a single edge. In the first case, this corresponds to partitioning the graph into a disjoint collection of cliques, while the second case corresponds to a disjoint collection of bicliques.

In both cases, the proof of correctness in [GR09] proceeds by assuming that the algorithm accepts with probability at least $\frac{1}{3}$ and then constructs a partitioning with few incorrect edges.

We now give an alternate proof for the case that H is a self-loop. Rather than show how to construct a good partitioning from the assumption that the graph was accepted with high probability, we instead directly show that if G is ϵ-far from $\mathcal{BUC}(H)$, then G is rejected with high probability.

3.1 The Adaptive Algorithm

Informally, the algorithm begins by guessing the distribution of edits. It then finds a vertex with many adjacent edits. Finally, it queries a small neighborhood of that vertex in order to verify the proposed witness. Although the algorithm stated here uses polylogarithmically fewer queries than the one given in [GR09], it is best viewed as a slight rephrasing of the one given in [GR09].

Algorithm 24. CLIQUECOLLECTION($G; \epsilon$)

```
for $i = 1$ to $\lg \frac{1}{\epsilon} + O(1)$ do
```

```
2:     for $O(2^i)$ trials do
3:         Select $u$ at random.
4:         Select $S$ at random, with $|S| = O(\frac{1}{\epsilon}\lg\frac{1}{\epsilon}/2^i)$.
5:         Query $(u, s)$ for all $s \in S$.
6:         Select $T \subset \Gamma_S(u)$, with $|T| = O(1)$.
7:         Query $T \times T$, and REJECT if any queries return 0.
8:         Query $T \times S \backslash T$, and REJECT if any queries return 1.
9:     end for
10: end for
11: ACCEPT.
```

The query complexity is clearly $O(\frac{1}{\epsilon}\lg^2\frac{1}{\epsilon})$.

The algorithm clearly accepts all $G \in \mathcal{CC}$. It remains to show that it rejects with probability at least 2/3 whenever G is ϵ-far from $\mathcal{CC}$.

Theorem 25. *If G is ϵ-far from $\mathcal{CC}$, then* CLIQUECOLLECTION *rejects with probability at least* 2/3.

To prove Theorem 25, we will frequently refer to the number of *edits* adjacent to a vertex. Formally, letting $\widetilde{G} = \arg\min_{H \in \mathcal{CC}} d(G, H)$, we define the number of edits to be $\left|\Gamma(u) \bigtriangleup \widetilde{\Gamma}(u)\right|$.

The key lemma for proving Theorem 25 is the following.

Lemma 26. *Let $\gamma_i = \frac{2^i\epsilon}{\lg\frac{1}{\epsilon}}$. If the vertex u selected in step (3) has $\gamma_i N \leq \left|\Gamma(u) \bigtriangleup \widetilde{\Gamma}(u)\right| \leq 2\gamma_i N$, then* CLIQUECOLLECTION *rejects with probability at least* 1/3.

Proof of Theorem 25. Note that since G is ϵ-far from $\mathcal{CC}$, there exists p, γ, with $\epsilon \leq p, \gamma \leq 1$ and $p\gamma \geq \epsilon/\lg\frac{1}{\epsilon}$, such that at least pN vertices are adjacent to between γN and $2\gamma N$ edits.

Consider the iteration of the outer loop in which $2^i \approx p$. With high probability, $\Omega(1)$ vertices are selected in step (3) which are adjacent to between γN and $2\gamma N$ vertices. Therefore, by Lemma 26, CLIQUECOLLECTION rejects with probability at least $1 - (\frac{2}{3})^{\Omega(1)} > 2/3$. $\square$

3.2 Proof Of Lemma 26

The proof is essentially a case analysis of the ways in which G can be ϵ-far from $\mathcal{BUC}(H)$.

Given a graph G, let $\mathcal{G} = \arg\min_{H \in \mathcal{CC}} d(G, H)$ be the clique collection closest to G. By a slight abuse of notation, we refer to the induced V^i as the "cliques" of G. We will generally refer to the V^i without explicit reference to $\mathcal{G}$.[1]

[1] We assume that ties are broken in a canonical way, so this is well-defined.

For ease of notation, we let $\Gamma_i(u) = V_i \cap \Gamma(u)$, $\overline{\Gamma_i}(v) = V^i \backslash \Gamma(u)$, and $\Gamma'(u) = \Gamma(u) \backslash V^i$, where $u \in V^i$.

Since $\mathcal{G}$ minimizes $d(G, \mathcal{G})$, we have the following observations.

Observation 27. *For all* $\Gamma \subset V^i$, $\left|E(\Gamma, V^i \backslash \Gamma)\right| \geq \frac{1}{2}\left|\Gamma\right|\left|V^i \backslash \Gamma\right|$.

Observation 28. *For all* $i \neq j$, $\left|E(V^i, V^j)\right| \geq \frac{1}{2}\left|V^i\right|\left|V^j\right|$.

For Observation 27, we could otherwise reduce the distance by splitting V^i. For Observation 28, we could otherwise reduce the distance by joining V^i and V^j.

Lemma 29. *Given* $\Gamma = \{v_1, \ldots, v_n\}$ *such that* $\left|V^i\right| < n/6$ *for all* i *such that* $\Gamma \cap V^i \neq \emptyset$, $\Pr[(v_j, v_k) \in E] < \frac{3}{4}$.

Proof. Let $p = \Pr[(v_j, v_k) \in E]$, and let $n_i = \left|V^i \cap \Gamma\right|$. Then the cost of moving Γ into its own clique is bounded above by

$$\left(\sum_i n_i \left|V^i\right| - \tbinom{n_i}{2}\right) + (1-p)\tbinom{n}{2} - \left(p\tbinom{n}{2} - \sum_i \tbinom{n_i}{2}\right)$$

which is bounded by

$$n \cdot \frac{n}{6} + (1-p)\tbinom{n}{2} - p\tbinom{n}{2} < \tfrac{2}{3}n^2 - pn^2 + pn$$

This is negative for $p \geq \frac{3}{4}$ and sufficiently large n. □

We are now ready to prove the lemma.

Let u be a vertex selected in step 3 which is adjacent to between γN and $2\gamma N$ edits. Note that since $|S| = \Omega(1/\gamma)$, with high probability S contains $\Omega(1)$ vertices v such that (u, v) is an incorrect edge (ie, is in $\Gamma(u) \bigtriangleup \widetilde{\Gamma(u)}$).

Let V^i be the clique containing u.

Case 1: $\left|\overline{\Gamma_i}(u)\right| \geq |\Gamma'(u)|$

Since $|\Gamma_i(u)| \geq \left|\overline{\Gamma_i}(u)\right|$, it follows that, with high probability, $|\Gamma_i(u) \cap \Gamma_S(u)| \geq \frac{1}{4}|\Gamma_S(u)|$. Furthermore, $|\Gamma_S(u)| = \Omega(1)$, since $|\Gamma_i(u)| > \frac{\gamma}{2}N$. Therefore, with high probability, T contains $\Omega(1)$ vertices in V^i.

Since $\left|\overline{\Gamma_i}(u)\right| \geq \frac{\gamma}{2}N$, $\left|\overline{\Gamma_i}(u) \cap S\right| = \Omega(1)$, with high probability.

We therefore have that step (8) contains $\Omega(1)$ queries between $\Gamma_i(u)$ and $\overline{\Gamma_i}(u)$. By Observation 27, each of these queries causes the algorithm to reject with probability at least 1/2.

Case 2: $|V^i| \geq \frac{\gamma}{100}N$ **and** $\left|\overline{\Gamma_i}(u)\right| < |\Gamma'(u)|$

First, note that since $\left|V^i\right| > \frac{\gamma}{100}N$ and $|\Gamma'(u)| < 2\gamma N$, $|\Gamma_i(u)| \geq \frac{1}{200}|\Gamma'(u)|$. Therefore, with high probability, G contains $\Omega(1)$ vertices in V^i.

Next, note that with high probability, there exists $v \in \Gamma_S(u) \cap V^j$, for some $j \neq i$.

We have multiple cases, depending on the values of $\Gamma_i(u)$, $\Gamma_i(v)$, $\Gamma_j(u)$, and $\Gamma_j(v)$. We first note that if $\Gamma_i(v) > 0.9n_i$, then $\left|V^j\right| \geq 0.8\left|V^i\right|$. For if not, the cost of moving v to V^i would be upper-bounded by $0.1n_i + n_j - 0.9n_i < 0.8n_i - 0.8n_i = 0$.

Case 2a: $\Gamma_i(u) < 0.95\left|V^i\right|$

Then $\left|\overline{\Gamma_i}(u) \cap S\right| = \Omega(1)$, with high probability. We therefore make $\Omega(1)$ queries between $\Gamma_i(u)$ and $\overline{\Gamma_i}(u)$ in step (7). By Observation 27, each of these causes the algorithm to reject with probability at least 1/2.

Case 2b: $\Gamma_i(u) > 0.95\left|V^i\right|$ **and** $\Gamma_i(v) < 0.9\left|V^i\right|$

Then $\left|\Gamma_i(u) \cap \overline{\Gamma_i}(v) \cap S\right| = \Omega(1)$, with high probability. We therefore reject in step (7) with certainty.

Case 2c: $\Gamma_i(u) > 0.95\left|V^i\right|$**,** $\Gamma_i(v) > 0.9\left|V^i\right|$**, and** $\Gamma_j(u) < 0.75\left|V^j\right|$

First, note that $\left|V^j\right| > 0.8\left|V^i\right|$, for otherwise we would move v into V^i. Therefore, with high probability, S contains a vertex $w \in \overline{\Gamma_j}(u)$.

Since v was uniformly chosen from $\Gamma_j(u)$, $\Pr[(v,w) \in E] \geq \frac{1}{2}$, by Observation 27.

Case 2d: $\Gamma_i(u) > 0.95\left|V^i\right|$**,** $\Gamma_i(v) > 0.9\left|V^i\right|$**, and** $\Gamma_j(v) \geq 0.6\left|V^j\right|$

With high probability, S contains $\Omega(1)$ vertices in $\Gamma_j(v)$. We therefore query $\Omega(1)$ random edges between $0.95V^i$ and $0.6V^j$ in step (7). But $E(V^i, V^j) \leq \frac{1}{2}\left|V^i\right|\left|V^j\right|$, so each such query cause a rejection with probability at least

$$\frac{0.95\left|V^i\right| \cdot 0.6\left|V^j\right| - \frac{1}{2}\left|V^i\right|\left|V^j\right|}{0.95\left|V^i\right| \cdot 0.6\left|V^j\right|} \approx 0.12$$

Case 2e: $\Gamma_i(u) > 0.95\left|V^i\right|$**,** $\Gamma_i(v) > 0.9\left|V^i\right|$**,** $\Gamma_j(u) > 0.75\left|V^j\right|$ **and** $\Gamma_j(v) < 0.6\left|V^j\right|$
Since $\left|\Gamma_j(u) \cap \overline{\Gamma_j(v)}\right| > 0.15\left|V^j\right| \geq \frac{\gamma}{25}N$, with high probability S contains $\Omega(1)$ such vertices. Each of these causes the algorithm to reject with certainty.[2]

Case 3: $|V^i| < \frac{\gamma}{100}N$**,** $\left|\overline{\Gamma}_i(u)\right| < |\Gamma'(u)|$

We have two subcases, depending on whether $\Gamma'(u)$ is primarily contained in large or small cliques. Given $v \in \Gamma'(u)$, let $V^j \ni v$.

Case 3a: $\Pr[\left|V^j\right| \geq \frac{\gamma}{24}N] \geq 1/2$
With high probability, T contains a vertex $v \in V^j$ such that $\left|V^j\right| \geq \frac{\gamma}{24}N$.

Note that $\left|\overline{\Gamma_j}(u)\right| > \frac{\gamma}{100}N$. So with high probability, there exists $w \in S \cap \overline{\Gamma_j}(u)$. By Observation 27, $\Pr[(v, w) \in E] \geq \frac{1}{2}$, so the algorithm rejects in step (8) with high probability.

Case 3b: $\Pr[\left|V^j\right| \geq \frac{\gamma}{24}N] < 1/2$

Let $\widetilde{\Gamma} = \{v \in \Gamma'(u) \,:\, \left|V^j\right| > \frac{\gamma}{8}N\}$. By assumption, $\left|\widetilde{\Gamma}\right| \geq \frac{\gamma}{4}N$. With high probability, T contains $\Omega(1)$ vertices in $\widetilde{\Gamma}$. Therefore, in step (7), we makes $\Omega(1)$ queries from $\widetilde{\Gamma} \times \widetilde{\Gamma}$, each of which causes the algorithm to reject with probability at least 1/4, by Lemma 29. ■

[2] Case 2e cannot actually happen. For if $\Gamma_i(v) > 0.9n_i$ and $\Gamma_j(v) < 0.6n_j$, then the cost of moving v to V^i is bounded by $0.1n_i + 0.6n_j - 0.9n_i - 0.4n_j = 0.2n_j - 0.8n_i$, which means that $n_j > 4n_i$. But then the cost of moving u to V^j is bounded by $n_i + 0.25n_j - 0.75n_j < 0.25n_j + 0.25n_j - 0.75n_j < 0$.

Chapter 4

Towards Proving an Upper Bound on the Adaptive Query Complexity of $\mathcal{BUC}(H)$ for Arbitrary H

Although the results given in [GR09] and Chapter 2 provide strong evidence that the canonical transformation is essentially optimal, proving that result for properties which are neither promise problems nor proximity-dependent remains open. In this section, we restate the main structural lemmas of [GR09], used in that paper for proving the query complexity of $\mathcal{BUC}(H)$ when H is a single loop or single edge, in the context of general H. Although these proofs are essentially direct adaptations of those given in [GR09], we feel that formally stating the general case yields additional insight as to the key combinatorial difficulties.

The two key concepts needed to discuss blow-up collections of an arbitrary base graph are *skeletons* and *inconsistent edges*. Informally, a skeleton is a set of vertices which are believed to belong to different components of the blow-up collection. An inconsistent edge, relative to a skeleton, is an edge which contradicts the proposed decomposition.

Definition 30 (Skeleton)**.** *A set $T \subseteq V$ is a* skeleton *if $\Gamma_T(u) \neq \Gamma_T(v)$ for all $u, v \in T$. Given $S \subseteq V$, skeleton$(S) \subseteq S$ is the largest skeleton contained in S.*

Note that a skeleton of size $|H|+1$ is a witness against $\mathcal{BU}(H)$ and a connected skeleton is a witness against $\mathcal{BUC}(H)$. Also note that if S is connected, *skeleton*(S) is connected.

Definition 31 (Inconsistent Edge)**.** *Fix $T \subseteq V$. An edge (v, w) is* inconsistent *with T if $\Gamma_{T'}(v) \neq \Gamma_{T'}(u)$ for all $u \in T$, where $T' = T \cup \{v, w\}$.*

Given these two definitions, we conjecture that the following algorithm, informally described in [GR09] for the case that H is a simple cycle, correctly tests $\mathcal{BUC}(H)$ for all H.

Algorithm 32. SubTest$(u; \delta)$

```
1:  Let $T = \{u\}$.
2:  while $|T| \leq |H|$ do
3:      Choose $O(\frac{1}{\delta})$ vertices $S$.
4:      Query $T \times S$.
5:      Choose $O(\frac{1}{\delta})$ edges uniformly from $\Gamma_S(T) \times \Gamma_S(T)$ and query them.
6:      Choose $O(\frac{1}{\delta})$ edges uniformly from $\Gamma_S(T) \times S \backslash \Gamma_S(T)$ and query them.
7:      If an inconsistent edge $(v, w)$ was found in steps 4-6, $T \leftarrow$ skeleton$(T \cup \{v, w\})$.
8:      Else break.
9:  end while
10: If $G|_T \not\subseteq H$, REJECT. Otherwise, ACCEPT.
```

Algorithm 33. BlowUpCollection$(G; \epsilon)$

```
for $i = 1$ to $\lg \frac{1}{\epsilon} + O(1)$ do
    Run SubTest$(u; 2^i \epsilon)$ for $O(2^i)$ randomly chosen $u$.
end for
Accept if all calls to SubTest accepted. Reject if any call rejected.
```

4.1 Basic Behavior of SubTest

Before stating the key lemmas, we formally define the two ways in which a graph can fail to be a valid blow-up collection.

Definition 34. *Given $C \subseteq V$, let $\mathcal{E}_I(C) = d(G|_C, \mathcal{BUC}(H))$ be the number of edits required to make C a valid blow-up collection of H, and let $\mathcal{E}_E(C) = |E(C, V \backslash C|$ be the number of edits needed to disconnect C from the rest of G. Let $\mathcal{E}(C) = \mathcal{E}_I(C) + \mathcal{E}_E(C)$.*

Given a skeleton $T \subseteq C$ such that $C = \Gamma(T)$, note that $\mathcal{E}(C)$ is at most the number of edges in C that are inconsistent with T.

Given a skeleton T, we say that T is *terminal* if SubTest terminates on T. Unless otherwise specified, all skeletons are assumed to be terminal.

Let $C_T = \{u \in \Gamma(T) : |\Gamma(u) \backslash \Gamma(T)| < \frac{1}{5} |\Gamma(T)|\}$.

The following 3 lemmas and their proofs are essentially restatements of the proofs in [GR09], Claims 3.2.2 and 3.2.4 (and the surrounding discussion).

Lemma 35. *Fix T such that $|\Gamma(T)| > \delta N$ and steps 4-6 fail to find any inconsistent edges with probability $1/3$. Then $|\Gamma(T) \backslash C_T| < \frac{\delta}{10} N$.*

Proof. Suppose otherwise. Then there are $\alpha |\Gamma(T)|$ vertices connected to βN external edges, where $\alpha = \frac{\delta N}{10|\Gamma(T)|}$ and $\beta = \frac{|\Gamma(T)|}{10N}$.

By a Chernoff bound, S contains at least

$$\frac{1}{2} \cdot \frac{\alpha |\Gamma(T)|}{N} \cdot \frac{c}{\delta} > \frac{c}{8}$$

such vertices with high probability. Consider such a high-degree vertex u. By a Chernoff bound, $|\Gamma(u) \cap S| > \frac{\beta}{2} \frac{c}{\delta}$ with high probability

Therefore, with high probability, $(\Gamma(T) \cap S) \times (S \backslash \Gamma(T))$ contains at least

$$\frac{\alpha}{2} |\Gamma(T) \cap S| \cdot \frac{\beta}{2} |S \backslash \Gamma(T)| > \frac{\delta}{16} |\Gamma(T) \cap S| \cdot |S \backslash \Gamma(T)|$$

edges. So with high probability, step 6 finds such an edge. □

Lemma 36. *Fix T such that $|\Gamma(T)| > \delta N$ and steps 4-6 fail to find any inconsistent edges with probability $1/3$. Then $\mathcal{E}_E(C_T) < \delta |C_T| N$.*

Proof. By lemma 35, $|E(C_T, \Gamma(T) \backslash C_T)| \leq |C_T| |\Gamma(T) \backslash C_T| < \frac{\delta}{10} N |C_T|$.

We claim that $|E(C_T, V \backslash \Gamma(T))| < \frac{\delta}{2} N |C_T|$.

Suppose otherwise. Then there must be at least $\alpha |\Gamma(T)|$ vertices in C_T which are connected to at least βN vertices in $V \backslash \Gamma(T)$, where $\alpha\beta = \frac{\delta}{4}$. Note that since $\beta N < \frac{1}{5} |\Gamma(T)|$, $\alpha > \frac{\delta N}{4|\Gamma(T)|}$. Therefore, by the same calculation as in the proof of lemma 35, with high probability step 6 finds such an edge. □

Lemma 37. *Fix T such that $|\Gamma(T)| > \delta N$ and steps 4-6 fail to find any inconsistent edges with probability $1/3$. Then $\mathcal{E}_I(C_T) < \delta |C_T| N$.*

Proof. Suppose otherwise, and let

$$\rho = \frac{\delta |C_T| N}{|\Gamma(T)|} > \frac{\delta |\Gamma(T)| N}{2 |\Gamma(T)|^2} = \frac{\delta N}{|\Gamma(T)|}$$

denote the lower bound on the fraction of inconsistent edges in $\Gamma(T) \times \Gamma(T)$.

Clearly, $\mathbb{E}[|\Gamma_S(T)|] = \frac{|\Gamma(T)|}{N} \cdot \frac{c}{\delta}$, and so with high probability, $|\Gamma_S(T)| > \frac{|\Gamma(T)|}{N} \cdot \frac{c}{2\delta}$. We assume that this is the case, and let $n = |\Gamma_S(T)|$. Note that $|\Gamma_S(T)| > \frac{c}{2}$.

For each pair of vertices $v_i, v_j \in \Gamma_S(T)$, define an indicator variable ζ_{ij} which is 1 if (v_i, v_j) is inconsistent with T. By assumption, $\mathbb{E}[\zeta_{ij}] \geq \rho$. We now bound $\mathbb{V}[\sum \zeta_{ij}] / \mathbb{E}[\sum \zeta_{ij}]^2$; it then follows that the fraction of inconsistent edges in $\Gamma_S(T) \times \Gamma_S(T)$ is at least $\rho/2$ with high probability, and so step 5 finds such an edge with high probability.

Note that ζ_{ij} and $\zeta_{i'j'}$ are independent if $\{v_i, v_j\} \cap \{v_{i'}, v_{j'}\} = \emptyset$. Let $\widetilde{\rho} = \mathbb{E}[\zeta_{ij}]$. We have

$$\mathbb{E}\left[\sum_{i,j} \zeta_{ij}\right] = \binom{n}{2}\widetilde{rho} > \frac{n^2\widetilde{\rho}}{3}$$

and

$$\begin{aligned}
\mathbb{V}\left[\sum_{ij} \zeta_{ij}\right] &= \mathbb{E}\left[\left(\sum_{i,j} \zeta_{ij}\right)^2\right] - \mathbb{E}\left[\sum_{i,j} \zeta_{ij}\right]^2 \\
&= \mathbb{E}\left[\sum_{i,j,i',j'} \zeta_{ij}\zeta_{i'j'}\right] - \sum_{iji'j'} \mathbb{E}[\zeta_{ij}]\,\mathbb{E}[\zeta_{i'j'}] \\
&= \sum_{\substack{i,j,i',j' \\ \text{disjoint}}} \mathbb{E}[\zeta_{ij}]\,\mathbb{E}[\zeta_{i'j'}] + \sum_{\substack{i,j,i',j' \\ \text{otherwise}}} \mathbb{E}[\zeta_{ij}\zeta_{i'j'}] - \sum_{iji'j'} \mathbb{E}[\zeta_{ij}]\,\mathbb{E}[\zeta_{i'j'}] \\
&< \sum_{\substack{i,j,i',j' \\ \text{disjoint}}} \mathbb{E}[\zeta_{ij}\zeta_{i'j'}] < 4\sum_{i,j,j'} \mathbb{E}[\zeta_{ij}\zeta_{ij'}] \\
&\leq 4\sum_{i,j,j'} \mathbb{E}[\zeta_{ij}] = 4n\sum_{i,j} \mathbb{E}[\zeta_{ij}] \\
&= 4n\binom{n}{2}\widetilde{\rho} < 2n^3\widetilde{\rho}
\end{aligned}$$

Combining these bounds, and the fact that $\widetilde{\rho} \geq \rho$, we get

$$\frac{\mathbb{V}[\sum_{i,j} \zeta_{ij}]}{\mathbb{E}[\sum_{i,j} \zeta_{ij}]^2} < \frac{2n^3\widetilde{\rho}}{(n^2\widetilde{\rho}/3)^2} < \frac{18}{n\widetilde{\rho}} \leq \frac{18}{n\rho} < 18\frac{|\Gamma(T)|}{\delta N}\frac{2\delta N}{c\,|\Gamma(T)|} = \frac{36}{c}$$

which can be made arbitrarily small. □

We now combine the previous three lemmas to show that if SubTest$(v;\delta)$ accepts with reasonable probability, then we can construct a component containing v which is close to being an isolated blow-up.

Corollary 38. *If* SubTest$(v;\delta)$ *accepts with probability at least* $\frac{1}{3}$*, then there exists* $C \ni v$ *such that* $\mathcal{E}(C) < 2\delta N \cdot |C|$.

Proof. If $|\Gamma(v)| < \delta N$, then let $C = \{v\}$.

Otherwise, let T be a skeleton containing v such that SubTest terminates with high probability. Note that such a T must exist, since otherwise SubTest would have rejected after $|H|+1$ iterations. Set $C = C_T$.

By lemmas 37 and 36, $\mathcal{E}(C_T) = \mathcal{E}_I(C_T) + \mathcal{E}_E(C_T) < 2\delta\,|C_T|\,N$, as desired. □

4.2 Sketch of the Reconstruction Procedure

Given a skeleton T, we say that T is *δ-good* if steps 4-6 fail to find an inconsistent edge and T is consistent with H. Similarly, we say that T is not δ-good if T is not consistent with H. Otherwise, we recursively say that T is δ-good if and only if, with high probability over the choice of inconsistent edge (v, w) in step 7, *skeleton*$(T \cup \{v, w\})$ is δ-good.

Call a vertex u δ-good if $\{u\}$ is δ-good and $|\Gamma(u)| > \delta N$.

Lemma 39. *If algorithm 33 accepts with probability at least* $1/3$*, then there are at most* $2^{-i}N$ *vertices which are not* $2^i\epsilon$*-good and have degree at least* $2^i\epsilon N$*, for each* $i < \lg \epsilon^{-1}$.

The lemma follows immediately from the definition of i-goodness.

Informally, we now show how to construct a blow-up collection close to G. The procedure progresses in rounds. In the initial round, while there remains an uncovered vertex v which is ϵ-good, let C be the component guaranteed by lemma 38 and “fix” that component. Once all ϵ-good vertices have been covered, disconnect all uncovered low-degree vertices and then go on to the next round.

In general, during the i-th round, we fix any remaining $2^i\epsilon$-correct vertices and then disconnect any leftover low-degree vertices.

Suppose that the components constructed in this procedure were actually disjoint. Then the total cost of the procedure would be bounded by

$$\sum_i \sum_C 2^i \epsilon \, |C| \, N < \sum_i 2^i \epsilon \cdot 2^{-i} N \cdot N = \epsilon N^2$$

since at most 2^i vertices are incorrect at the start of the i-th round.

All that remains, therefore, is to show that we can select the components to be essentially disjoint (ie, such that $|\bigcup C| \approx \sum |C|$), while still covering most of the i-good vertices in each round. Specifically, we will require that each additional component constructed by the reconstruction procedure contain a constant fraction of new vertices. The challenge will then be to ensure that enough vertices are covered during each round.

Since the reconstruction will not yield a strict partitioning, the following observation will be needed.

Lemma 40. *Let $C = \bigcup_i C_i$, with the C_is not necessarily disjoint. Then $\mathcal{E}(C) < \sum \mathcal{E}(C_i)$.*

Proof. Consider the partitioning of C obtained by assigning each $v \in C$ arbitrarily to some C_i which contains it. Within each partition $\widetilde{C_i} \subseteq C_i$, use the structure inherited from C_i.

Note that $\sum \mathcal{E}_I(\widetilde{C_i}) \leq \sum \mathcal{E}_I(C_i)$. Similarly, $\sum \mathcal{E}_E(\widetilde{C_i}) \leq \sum \mathcal{E}_E(C_i)$.

Since any partitioning provides an upper bound on $\mathcal{E}(C)$, it follows that $\mathcal{E}(C) \leq \sum \mathcal{E}_I(C_i) + \mathcal{E}_E(C_i) = \sum \mathcal{E}(C_i)$, as desired. □

4.3 The Reconstruction Procedure

As stated in the previous discussion, the reconstruction proceeds in rounds $i = 1 \ldots, \lg \frac{1}{\epsilon}$.

Throughout this section, we informally let C be the set of vertices which have been "covered" by the reconstruction, F_i be the set of skeletons used in round i, R_i be the set of vertices which are not yet covered at the start of round i, L_i be the set of vertices discarded as "low-degree" at the end of round i, and $L = \bigcup_{j<i} L_j$ be the vertices which have been discarded as "low-degree" so far. These will be formally defined later.

4.3.1 Cliques

We begin with a discussion of the simplest case, in which H is a single loop. This proof was given in [GR09], but we give it here for completeness, and for comparison to the general case.

1. Pick an arbitrary vertex $v \in R_{i-1} \backslash C$ such that
   (a) v is δ-good
   (b) $|\Gamma(v)| > \delta N$
   (c) $|\Gamma(v) \cap (C \cup L)| \leq \frac{1}{10} |\Gamma(v)|$
2. For a vertex v selected in step 1, update $F_i \leftarrow F_i \cup \{T\}$ and $C \leftarrow C \cup C_v$.
3. If no such vertices remain, set $L_i = \{v \in R_{i-1} \backslash C : |\Gamma(v)| < \delta N\}$ and $R_i = R_{i-1} \backslash (C \cup L_i)$. Update $L \leftarrow L \cup L_i$.

We now show that this yields a good partitioning of most remaining vertices.

Lemma 41. *Suppose* $|R_{i-1}| < 2^{-i-1}N$. *Then* $\sum_{v \in F_i} \mathcal{E}(C_v) < \epsilon N^2$.

Proof. Consider $v \in F_i$. Since $|\Gamma(v)| > \delta N$ and $|C_v \cap (C \cup L)| < \frac{1}{10} |\Gamma(v)|$ (by condition (1c)), $|C_v \cap (C \cup L)| < \frac{1}{10} |\Gamma(v)|$. By lemma 35, it follows that C_v is at least half "new" vertices.

Therefore, $\sum_{v \in F_i} |C_v| < 2 |R_i|$, and the result follows from corollary 38. □

We now show that, despite condition (1c), most high-degree vertices are still covered.

Lemma 42. *For all* i, $\left|R^i\right| < 2^{-i}N$.

Proof. We proceed by induction on i. For $i = 0$, the claim is trivial.

Any vertex in R_i is either not δ-good, or has $|\Gamma(v)| > \delta N$ and $|\Gamma(v) \cap (C \cup L)| > \frac{1}{10} |\Gamma(v)| > \frac{\delta}{10} N$. By lemma 39, there are at most $\frac{1}{2} 2^{-i} N$ vertices which are not δ-good.

Let $\widetilde{R_i} \subset R_i$ be the set of δ-good vertices which remain at the end of round i, and consider some $v \in \widetilde{R_i}$. Since v is δ-good, at most $\frac{1}{2}$ of it's neighbors have degree less than δN, which means that it has at least $\frac{\delta}{20} N$ neighbors in C. The total number of edges between R_i and C is therefore at least $\left|\widetilde{R_i}\right| \frac{\delta}{10} N$. But by lemma 41, the total number of such edges is at most ϵN^2. So $\left|\widetilde{R_i}\right| < \frac{1}{2} 2^{-i} N$. □

The key simplification that occurs in this case, as opposed to general H, is that C_v has large overlap with $C \cup L$ if and only if C_v has large overlap with C.

4.3.2 General H

We now consider the general case. As with the case for cliques, we again use a more stringent selection criteria, and then attempt to argue that R_i remains small.

1. Pick an arbitrary vertex $v \in R_{i-1} \backslash C$ and $u_1, \ldots, u_k$ such that $u_j \in \Gamma(v \cup \bigcup_{\ell<j} u_\ell)$ such that
   (a) $T = \{v, u_1, \ldots, u_k\}$ is δ-good
   (b) $|\Gamma(v)| > \delta N$
   (c) $|\Gamma(v) \cap (C \cup L)| \leq \frac{1}{10 \cdot 2^H} |\Gamma(v)|$
   (d) $|\Gamma(T) \cap (C \cup L)| < \frac{1}{10 \cdot 2^{H-j}} |\Gamma(T)|$
2. For a skeleton T selected in step 1, update $F_i \leftarrow F_i \cup \{T\}$ and $C \leftarrow C \cup C_T$.
3. If no such skeletons remain, set $L_i = \{v \in R_{i-1} : |\Gamma(v)| < \delta N\}$ and $R_i = R_{i-1} \backslash (C \cup L_i)$. Update $L \leftarrow L \cup L_i$.

As in the case of cliques, it suffices to bound the size $|R_i|$, as shown by the following lemma.

Lemma 43. *Suppose* $|R_{i-1}| < 2^{-i-1} N$. *Then* $\sum_{T \in F_i} \mathcal{E}(C_T) < \epsilon N^2$.

Proof. Let $\delta = 2^i \epsilon$ and consider $T \in F_i$. Since $|\Gamma(T)| > \delta N$ and $|C_T \cap (C \cup L)| < \frac{1}{5} |\Gamma(T)|$ (by condition (1d)), $|C_T \cap (C \cup L)| < \frac{1}{5} |\Gamma(T)|$. By lemma 35, it follows that C_T is at least half "new" vertices.

Therefore, $\sum_{T \in F_i} |C_T| < 2 |R_i|$, and the result follows from corollary 38. □

Unfortunately, we do not know how to show that most high-degree vertices are still covered, despite conditions (1c) and (1d).

Conjecture 44. *For all* i, $\left|R^i\right| < 2^{-i} N$.

The key difficulty in proving Conjecture 44 is that a good vertex can have many neighbors which are low-degree. Specifically, we have that each δ-good vertex which has not been covered after round i

has at least δN edges into $C \cup L$. To obtain the bound on $|R_i|$, we therefore want to show that the total number of edges leaving $C \cup L$ is at most $\delta N \cdot 2^{-i} N = \epsilon N^2$.

For the special case where H is a clique, we argued that most of the edges between v and $C \cup L$ were actually between v and C. Since the number of edges leaving C is bounded by the number of edits allowed, and that tolerance is essentially a free parameter of the algorithm/reconstruction, we could force the upper bound to be as small as necessary. In this case, however, the edges could be from v to L. As a result, both the upper- and lower-bound depend on the degree bound used to define L.

Chapter 5

Conclusions and Open Problems

We have shown that there exist proximity-dependent graph properties for which a non-adaptive tester must suffer an almost-quadratic increase in its query complexity over an adaptive tester. This gives evidence that the canonical transformation is essentially optimal, in the worst case.

The primary open problem is to remove the proximity-dependence from the graph properties used in Theorem 3. In particular, for any $\delta > 0$, does there exist a *single* graph property Π such that testing Π requires $Q = \Omega(q^{2-\delta})$ queries for all $\epsilon > 0$? It also remains to determine whether there exists a nearly-quadratic separation when the adaptive algorithm is only allowed one-sided error. One approach to both of these questions is to prove an $\widetilde{O}(\epsilon^{-1})$ upper-bound for the adaptive query complexity of $\mathcal{BUC}(H)$ for general graphs, as sketched in chapter 4.

We also reiterate the intriguing question raised in [GR09] as to what relationships are possible between the adaptive and non-adaptive query complexities. Specifically, do there exists properties such that $Q = \widetilde{\Theta}(q^{2-\delta})$, with $\delta \neq \frac{2}{t}$? In particular, is it true that Q must either be $\widetilde{\Theta}(q)$ or $\widetilde{\Omega}(q^{4/3})$?

Finally, it remains to determine whether a gap exists between adaptive and non-adaptive property testers for properties with adaptive query complexity $\Omega(\epsilon^{-2})$. In particular, does there exist a gap when $q = \Omega(1)$ as a function of N? A plausible conjecture is that the non-adaptive query complexity is $Q = \widetilde{O}(\epsilon^{-1} \cdot q)$, or potentially even that the canonical query complexity is $\widetilde{Q} = \widetilde{O}(\epsilon^{-1} \cdot q)$. Unfortunately, we do not currently have any idea how to prove (or disprove) such a conjecture.

Bibliography

[AG11] Lidor Avigad and Oded Goldreich. Testing graph blow-up. In Leslie Goldberg, Klaus Jansen, R. Ravi, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 6845 of *Lecture Notes in Computer Science*, pages 389–399. Springer Berlin / Heidelberg, 2011.

[Alo01] N. Alon. Testing subgraphs in large graphs. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, FOCS '01, pages 434–, Washington, DC, USA, 2001. IEEE Computer Society.

[AS06] Noga Alon and Asaf Shapira. A characterization of easily testable induced subgraphs. *Comb. Probab. Comput.*, 15:791–805, November 2006.

[Avi09] Lidor Avigad. On the lowest level of query complexity in testing graph properties. Master's project, Weizmann Institute of Science, Department of Computer Science and Applied Mathematics, December 2009.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC '90, pages 73–83, New York, NY, USA, 1990. ACM.

[CRT05] Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM J. Comput.*, 34:1370–1379, June 2005.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45:653–750, July 1998.

[GKNR09] Oded Goldreich, Michael Krivelevich, Ilan Newman, and Eyal Rozenberg. Hierarchy theorems for property testing. In *Proceedings of the 12th International Workshop and*

13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX '09 / RANDOM '09, pages 504–519, Berlin, Heidelberg, 2009. Springer-Verlag.

[GR97] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 406–415, New York, NY, USA, 1997. ACM.

[GR00] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity*, 7, 2000.

[GR07] Mira Gonen and Dana Ron. On the benefits of adaptivity in property testing of dense graphs. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '07/RANDOM '07, pages 525–539, Berlin, Heidelberg, 2007. Springer-Verlag.

[GR09] Oded Goldreich and Dana Ron. Algorithmic aspects of property testing in the dense graphs model. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '09 / RANDOM '09, pages 520–533, Berlin, Heidelberg, 2009. Springer-Verlag.

[GT03] Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Struct. Algorithms*, 23:23–57, August 2003.

[KKR04] Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM J. Comput.*, 33:1441–1483, June 2004.

[PR02] Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Struct. Algorithms*, 20:165–183, March 2002.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials withapplications to program testing. *SIAM J. Comput.*, 25:252–271, February 1996.

[RS06] Sofya Raskhodnikova and Adam Smith. A note on adaptivity in testing properties of bounded degree graphs. *Electronic Colloquium on Computational Complexity*, 13, 2006.