Listing 6-Cycles in Sparse Graphs

Virginia Vassilevska Williams
MIT, Cambridge, MA, USA

Alek Westover
MIT, Cambridge, MS, USA

Abstract

This work considers the problem of output-sensitive listing of occurrences of $2k$-cycles for fixed constant $k \geq 2$ in an undirected host graph with m edges and t $2k$-cycles. Recent work of Jin and Xu (and independently Abboud, Khoury, Leibowitz, and Safier) [STOC 2023] gives an $O(m^{4/3} + t)$ time algorithm for listing 4-cycles, and recent work by Jin, Vassilevska Williams and Zhou [SOSA 2024] gives an $\widetilde{O}(n^2 + t)$ time algorithm for listing 6-cycles in n node graphs. We focus on resolving the next natural question: obtaining listing algorithms for 6-cycles in the sparse setting, i.e., in terms of m rather than n. Previously, the best known result here is the better of Jin, Vassilevska Williams and Zhou's $\widetilde{O}(n^2 + t)$ algorithm and Alon, Yuster and Zwick's $O(m^{5/3} + t)$ algorithm.

We give an algorithm for listing 6-cycles with running time $\widetilde{O}(m^{1.6} + t)$. Our algorithm is a natural extension of Dahlgaard, Knudsen and Stöckel's [STOC 2017] algorithm for detecting a $2k$-cycle. Our main technical contribution is the analysis of the algorithm which involves a type of "supersaturation" lemma relating the number of $2k$-cycles in a bipartite graph to the sizes of the parts in the bipartition and the number of edges. We also give a simplified analysis of Dahlgaard, Knudsen and Stöckel's $2k$-cycle detection algorithm (with a small polylogarithmic increase in the running time), which is helpful in analyzing our listing algorithm.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Graph algorithms, cycles listing, fine-grained complexity, sparse graphs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.92

Related Version *arXiv Version*: `https://arxiv.org/abs/2411.07499`

Supplementary Material *Software*: `https://github.com/awestover/listing-C6s-LP` [18] archived at `swh:1:dir:72ea68e559495cac21c615e3c3bfa8cf3a18653f`

Funding *Virginia Vassilevska Williams*: Supported by NSF Grant CCF-2330048, BSF Grant 2020356 and a Simons Investigator Award.

Acknowledgements We'd like to thank Nathan S. Sheffield, Claire Zhang, and Ce Jin for helpful discussions.

1 Introduction

Listing copies of a small pattern graph that occur as subgraphs of a host graph is a fundamental problem in algorithmic graph theory. In this work, we consider the problem of listing C_{2k}'s (i.e., $2k$-cycles) for fixed constant $k \geq 2$. Some examples of applications of C_{2k} listing include analyzing social networks [9], and understanding causal relationships in biological interaction graphs [14]. (See e.g. [19] for further motivation.) In the following discussion we consider an n vertex m edge graph G with t C_{2k}'s (where $k \geq 2$ will be clear from context and t is not necessarily known).

The main reason for focusing on even length cycles is that while there are dense graphs that contain no odd cycles (e.g. bipartite graphs), a classic result of Bondy and Simonovits [6] states that any graph with at least $100kn^{1+1/k}$ edges contains a C_{2k}, for any integer

$k \geq 2$. This fact enables efficient "combinatorial" algorithms for C_{2k} detection, in contrast to the case of odd length cycles where efficient C_{2k+1} detection algorithms are based on matrix multiplication. In fact, very simple reductions (e.g. [16]) show that for any $k \geq 1$, C_{2k+1} detection is at least as hard as triangle detection, and the latter problem is known to be fine-grained subcubically equivalent to Boolean Matrix Multiplication [17]. Thus, fast algorithms for odd cycles cannot avoid matrix multiplication. We focus on even cycles from now on.

A classic result of Yuster and Zwick [19] shows how to determine whether a graph contains a C_{2k} in $O(n^2)$ time for any given constant $k \geq 2$. This quadratic running time is conjectured to be optimal in general (see [2, 15, 8], also discussed below); in particular, [2] gave concrete evidence for the hardness of C_4 detection.

Nevertheless, in the regime of sparser graphs, where $m < o(n^{1+1/k})$ improvements are possible. For C_4's there is a simple $O(m^{4/3})$ algorithm [5] that matches the performance of the $O(n^2)$ algorithm of [19] for $m = \Theta(n^{3/2})$, and improves on the performance for all $m < o(n^{3/2})$. Dahlgaard, Knudsen and Stöckel [8] give an algorithm for C_{2k} detection with running time $O(m^{2k/(k+1)})$ for every constant $k \geq 2$. This matches the $O(n^2)$ running time from [19] for $m = \Theta(n^{1+1/k})$ and improves on it for $m < o(n^{1+1/k})$.

In fine-grained complexity, it is common to assume widely-believed hypotheses and use reductions to obtain conditional lower bounds for fundamental problems. For the special case of cycle detection problems, most conditional lower bounds are based on hypotheses related to triangle detection. One of the most common hypotheses is that triangle detection in n-node graphs does not have a "combinatorial" [1] $O(n^{3-\varepsilon})$ time algorithm for $\varepsilon > 0$ (in the word-RAM model). Vassilevska W. and Williams [17] showed that this hypothesis is equivalent to the so called BMM Hypothesis that postulates that there is no $O(n^{3-\varepsilon})$ time combinatorial algorithm for multiplying two $n \times n$ *Boolean* matrices.

As mentioned earlier, it is easy to show that under the BMM hypothesis, detecting any odd cycle C_{2k+1} requires $n^{3-o(1)}$ time. Dahlgaard, Knudsen and Stöckel [8] gave lower bounds for combinatorial detection of even cycles of fixed length in *sparse* graphs. They show that there is no combinatorial algorithm for C_6 detection, or C_{2k} detection for any $k > 4$, with running time $O(m^{3/2-\varepsilon})$ for $\varepsilon > 0$. Lincoln and Vyas [15] give a similar conditional lower bound under a different hypothesis, extending the lower bounds for potentially non-combinatorial algorithms. Lincoln and Vyas show that, for large enough constant k, a C_{2k} detection algorithm in graphs with $m \leq O(n)$ with running time $m^{3/2-\varepsilon}$ would imply an algorithm for max-3-SAT with running time $2^{(1-\varepsilon')n}n^{O(1)}$.

The problem of *listing* even cycles is less well-understood. Without improving the C_{2k} detection algorithms discussed above, the best result that we could hope for in general is an algorithm with running time $O(n^2 + t)$, where t is the number of $2k$-cycles in the graph, and $O(m^{2k/(k+1)} + t)$ in the sparse setting where $m < o(n^{1+1/k})$. Recently, [3] and [12] gave such an algorithm for C_4 listing. In fact, Jin and Xu's [12] algorithm gives a stronger guarantee: After $O(m^{4/3})$ pre-processing time, they can (deterministically) enumerate 4-cycles with $O(1)$ delay per cycle. Jin and Xu [12] (and concurrently by Abboud, Bringmann, and Fischer [1]) shows that, under the 3SUM Hypothesis, there is no algorithm for C_4 enumeration with $O(n^{2-\varepsilon})$ or $O(m^{4/3-\varepsilon})$ pre-processing time and $n^{o(1)}$ delay, for $\varepsilon > 0$.

[1] The class of combinatorial algorithms is not well defined, but intuitively refers to algorithms that do not use fast matrix multiplication as a subroutine.

For C_6's, Jin, Zhou and Vassilevska Williams [11] give an $\widetilde{O}(n^2+t)$ listing algorithm. For sparse listing algorithms, the previous state of the art is [5] whose work implies an $O(m^{(2k-1)/k}+t)$ time C_{2k} listing algorithm. See also [7] where the complexity of the harder problem of listing H-partite[2] subgraphs is investigated.

1.1 Our contributions

We consider the problem of listing all C_6s in an m-edge graph. The best known result so far is an $O(m^{5/3}+t)$ time algorithm that follows from the work of [5]. Our main result is the first improvement over this 27 year old running time:

▶ **Theorem 23.** *There is an algorithm for listing C_6's in time $\widetilde{O}(m^{1.6}+t)$.*

We now summarize the ideas needed to prove Theorem 23.

Theorem 23 follows easily after establishing a certain bound on the number of **capped k-walks** in a graph. Capped k-walks are a notion introduced by Dahlgaard, Knudsen and Stöckel in [8]. Roughly speaking, a capped k-walk is a walk of length k where the first vertex in the walk has higher degree than the remaining vertices in the walk (handling vertices of equal degree requires a bit of additional care). [8]'s $O(m^{2k/(k+1)})$ time C_{2k} detection algorithm is based on the following fact:

▶ **Theorem 4** ([8]). *Let G be a C_{2k}-free graph with maximum degree $\Delta(G) \leq m^{2/(k+1)}$. Then, there are at most $\widetilde{O}(m^{2k/(k+1)})$ capped k-walks in G.*

One of our main contributions is a *simplified proof* of Theorem 4, although with polylogarithmically worse guarantees than [8]. This simplified analysis of capped k-walks is quite helpful in obtaining Theorem 23. The key lemma used to prove Theorem 4 is a generalization to bipartite graphs of Bondy and Simonovits' classic theorem on the extremal number of C_{2k}'s. Specifically, the result is:

▶ **Theorem 1** ([8]). *Let G be a bipartite graph with vertex parts of sizes L, R and with m edges. If $m > 100k(L+R+(LR)^{(k+1)/(2k)})$ then G contains a C_{2k}.*

In order to use capped k-walks for a listing algorithm, it would be useful to have a **supersaturation** variant of Theorem 1, which would guarantee the presence of *many* C_{2k}'s if the edge density is large; the fact that this supersaturation result could help with listing was communicated to us by Jin and Zhou [13]. The supersaturation analog of Bondy and Simonovits' [6] extremal number for C_{2k}'s is known:

▶ **Theorem 2** ([10]). *For every integer $k \geq 2$, there exist constants c, C such that if G is an n-node graph with $m \geq Cn^{1+1/k}$ edges, then G contains at least $c(m/n)^{2k}$ copies of C_{2k}.*

Jin and Zhou [13] formulated the following conjectured generalization of Theorem 2:

▶ **Conjecture 25.** (The Unbalanced Supersaturation Conjecture [Jin and Zhou'24]) *Let G be an m edge bipartite graph with vertex bipartition $A \sqcup B$, with t C_{2k}'s. Suppose $m \geq 100k(|A|+|B|+(|A||B|)^{(k+1)/2k})$. Then,*

$$t \geq \Omega\left(\frac{m^{2k}}{|A|^k|B|^k}\right).$$

[2] A k-node H is an H-partite subgraph of a k-partite graph G with vertex set $V = \cup_{a\in V(H)} V_a$ if there are k vertices $v_1, \ldots, v_k$ such that $v_a \in V_a$ for each $a \in \{1, \ldots, k\}$ and the mapping $a \to v_a$ is an isomorphism between H and the subgraph of G induced by $v_1, \ldots, v_k$. In other words, instead of looking for an arbitrary subgraph of G isomorphic to H, one only focuses on the subgraphs with exactly one node in each V_a and such that the node picked from V_a corresponds to node a of H.

Jin and Zhou obtained the following conclusion using the approach of [8]:

▶ **Fact 3** ([13]). If the Unbalanced Supersaturation Conjecture is true, then there is an $\widetilde{O}(t + m^{2k/(k+1)})$ time algorithm for $2k$-cycle listing for all $k \geq 2$.

In Corollary 29 we give a simple proof of Jin and Zhou's fact above. Thus, an approach to obtaining the conjectured optimal running time of $\widetilde{O}(t + m^{2k/(k+1)})$ for C_{2k}-listing would be to prove Conjecture 25. Unfortunately, establishing or refuting Conjecture 25 remains a challenging open problem.

Our main technical contribution is a proof of a weaker version of Conjecture 25 for $k = 3$. Then, we show that this partial progress towards Conjecture 25 can be used in our simplified method for analyzing capped k-walks to obtain a bound on the number of capped k-walks, and consequentially an improved C_6 listing algorithm in the sparse setting (namely, Theorem 23).

1.2 Open Questions

Proving or refuting Conjecture 25 is an important open question. It also is valuable to consider other avenues towards obtaining C_{2k} listing algorithms, especially in case Conjecture 25 turns out to be false. The listing algorithms of [11], [12], [3] use a variety of combinatorial insights that could potentially be generalized to larger k. It is also possible that a hybrid approach is productive: one can first use progress towards proving Conjecture 25 to force the instance to have a specific structure, and then use different combinatorial insights to solve the structured version of the problem.

1.3 Paper Outline

In Section 2 we present our simplified analysis of [8]'s C_{2k} detection algorithm. In Section A we show how to modify our simple analysis of [8]'s C_{2k} detection algorithm to get a C_{2k} listing algorithm, assuming the Unbalanced Supersaturation Conjecture. In Section 3 we present progress towards resolving the Unbalanced Supersaturation Conjecture. In Section 4 we show how to use our progress towards Conjecture 25 in our simplified capped k-walk analysis to obtain a listing algorithm for C_6's with running time $\widetilde{O}(m^{1.6} + t)$.

1.4 Notations

We use $\Delta(G)$ to denote the maximum degree of graph G. We write $N(v)$ to denote the neighborhood of vertex v, and for $S \subseteq V(G)$ we write $N_S(v)$ to denote $N(v) \cap S$. For graph G, we will use $V(G), E(G)$ to denote the vertex and edge sets of G. When G is clear from the context we also denote $V(G)$ by V and $E(G)$ by E. We let $n = |V|, m = |E|$ (when the graph being discussed is clear), and assume $m \geq n$. For vertex subsets A, B we write $e(A, B)$ to denote $|E \cap (A \times B)|$. A walk / path of length k will refer to a walk / path with k edges. Given $A, B \subseteq V$, we will write $G[A, B]$ to denote the induced subgraph on A, B, i.e., a graph with vertex set $A \cup B$ and edge set $E(G) \cap (A \times B)$. We will write $G[A]$ to denote $G[A, A]$.

We write $[x]$ to denote the set $\{1, 2, \ldots, \lceil x \rceil\}$. We use log to denote the base-2 logarithm. Define the **dyadic intervals** $\mathcal{I}_n$ as follows: letting n' denote the power of two in $[n, 2n)$, we define $\mathcal{I}_n = (\{0\}, \{1\}, [2, 4), [4, 8), \ldots, [n'/4, n'/2), [n'/2, n'])$. Note that $|\mathcal{I}_n| = 1 + \log n'$. For $j \in [\log n']$, the j-th dyadic interval refers to the j-th set in the list $\mathcal{I}_n$ of dyadic intervals, *starting counting from* $\{1\}$, or in other words *excluding the set* $\{0\}$. That is, the j-th dyadic interval is $[2^{j-1}, 2^j)$ for $j \in [(\log n') - 1]$ and the $(\log n')$-th one is $[n'/2, n']$.

2 A Simpler Analysis of the DKS C_{2k} Detection Algorithm

Dahlgaard, Knudsen and Stöckel [8] define **capped k-walks** and give a sophisticated analysis to bound the number of capped k-walks in C_{2k}-free graphs (assuming $\Delta(G) \leq m^{2/(k+1)}$). In this section we provide a simpler – although quantitatively worse by logarithmic factors – version of their analysis. This will be useful in future sections when we analyze the number of capped k-walks in graphs that are not C_{2k}-free.

Given a total ordering $\succ$ of the vertices in a graph, a **$\succ$-capped k-walk** [8] is a walk $x_0, x_1, \ldots, x_k$ where $x_0 \succ x_i$ for all $i \in [k]$. Throughout the paper we will assume implicitly an ordering $\succ$ of the vertices that satisfies $v_i \succ v_j$ whenever $\deg(v_i) > \deg(v_j)$ (but is otherwise arbitrary). Thus, we will refer simply to capped k-walks (leaving the dependence on $\succ$ implicit). The main result of this section is a simplified proof of the following fact:

▶ **Theorem 4** ([8]). *Let G be a C_{2k}-free graph with maximum degree $\Delta(G) \leq m^{2/(k+1)}$. Then, there are at most $\widetilde{O}(m^{2k/(k+1)})$ capped k-walks in G.*

We note that if G were (m/n)-regular then it would contain $\Theta(n(m/n)^k) \leq O(m^{2k/(k+1)})$ k-walks due to $m \leq O(n^{1+1/k})$ (which holds because G is C_{2k}-free). While a non-regular graph G may have more than $O(m^{2k/(k+1)})$ many k-walks, Theorem 4 states that G does not have more than this many *capped* k-walks (assuming $\Delta(G) \leq m^{2/(k+1)}$). At a high level, the proof of the bound on capped k-walks follows from applying the bipartite generalization of Bondy and Simonovits' extremal number bound [6] (Theorem 1) to a certain **layered organization** of the graph with nice regularity properties. This layered organization of the graph is our contribution.

In [8], the authors took a different approach: they defined the "ϕ-norm" of a vector v as

$$||v||_\phi = \int_0^\infty \sqrt{|\{i \mid |v_i| \geq x\}|}dx.$$

They then related the ϕ-norm of $X_G^k \mathbb{1}$, (where X_G is the graph's adjacency matrix and $\mathbb{1}$ is the all-ones vector) to the number of capped k-walks in G, and analyzed a large set of inequalities using combinatorial insights to bound this ϕ-norm. We find that the layered organization of the graph technique is more helpful than the ϕ-norm approach in elucidating [8]'s elegant result, and that this layered organization technique is easier to use in bounding the number of capped k-walks in graphs which are not C_{2k}-free. Our layered organization of the graph is depicted in Figure 1, and formally defined and analyzed in Lemma 5.

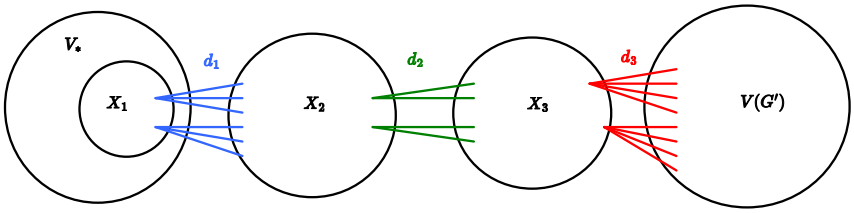


Figure 1 A layer pattern capturing many capped 3-walks.

▶ **Lemma 5.** *Fix graph G. There exists $d_* \in [n]$ such that if we let V_* denote the set of vertices with degree between $[d_*/2, d_*]$ and let G' denote the induced subgraph of G on vertices with degree at most d_*, then there exist $(X_1, \ldots, X_k) \subseteq V_* \times V(G')^{k-1}, (d_1, \ldots, d_k) \in \mathbb{N}^k$ such that letting $X_{k+1} = V(G')$,*

1. *The number of capped k-walks in G is at most $O(\log^{k+1} n)$ times the number of k-walks in $X_1 \times X_2 \times \cdots \times X_k \times V(G')$,*
2. *For each $i \in [k]$, every $v \in X_i$ has $|N_{X_{i+1}}(v)| \in [d_i/2, d_i]$.*

Proof. For $j \in [\log n]$, let V_j denote the set of vertices $v \in V$ with degree lying in the j-th dyadic interval. Fix j maximizing the number of capped k-walks in G that start in a vertex in V_j. Let $V_* = V_j, d_* = 2^j$, and let G' be the induced subgraph on vertices of degree at most d_*. The number of capped k-walks in G is at most $O(\log n)$ times the number of k-walks in G' that start in V_*; this is because the vertices in $V(G) \setminus V(G')$ are of too high degree to participate in a capped k-walk starting in V_*, and because every capped k-walk must start in some V_j.

For each $i_k \in [\log n]$ let $W^{(k)}_{i_k}$ denote the set of vertices in G' whose degree in G' lies in the i_k-th dyadic interval. Now, iteratively for $\ell = k-1, k-2, \ldots, 2$, for each $i_\ell \in [\log n]$ define $W^{(\ell)}_{i_k, i_{k-1}, \ldots, i_\ell}$ to be the set of vertices $v \in V(G')$ such that the number of neighbors of v in $W^{(\ell+1)}_{i_k, i_{k-1}, \ldots, i_{\ell+1}}$ lies in the i_ℓ-th dyadic interval. Finally, for each $i_1 \in [\log n]$ define $W^{(1)}_{i_k, i_{k-1}, \ldots, i_1}$ to be the set of vertices $v \in V_*$ such that the number of neighbors of v in $W^{(2)}_{i_k, i_{k-1}, \ldots, i_2}$ lies in the i_1-th dyadic interval.

Intuitively, looking at Figure 1, we are building up the sets X_ℓ from right to left. Here we think of $W^{(\ell)}_{i_k, i_{k-1}, \ldots, i_\ell}$ as representing X_ℓ for some choice of $i_k, i_{k-1}, \ldots, i_\ell$ that we will fix shortly.

Now, fix a choice $(i_k, \ldots, i_1) \in [\log n]^k$ that maximizes the number of k-walks

$$(x_1, \ldots, x_{k+1}) \in W^{(1)}_{i_k, \ldots, i_1} \times W^{(2)}_{i_k, \ldots, i_2} \times \ldots, W^{(k)}_{i_k} \times V(G'). \tag{1}$$

Note that every k-walk in G' starting in V_* must be of the form described in Equation (1) for *some* $(i_k, \ldots, i_1) \in [\log n]^k$. Thus, for our choice of $(i_k, \ldots, i_1)$, at least an $\Omega(1/\log^k n)$-fraction of the k-walks in G' starting in V_* are of the form Equation (1). For this choice of $(i_k, \ldots, i_1)$, we define for each $j \in [\log n]$,

$$X_j = W^{(j)}_{i_k, \ldots, i_j}, \quad d_j = 2^{i_j}.$$

These are the sets X_j, d_j that we wanted in the lemma statement, and they satisfy the required degree regularity property. Also because an $\Omega(1/\log^k n)$-fraction of the k-walks in G' starting in V_* are of the form Equation (1), and because the number of capped k-walks in G is at most $O(\log n)$ times the number of k-walks in G' starting in V_*, we have that the number of capped k-walks in G is at most $\widetilde{O}(1)$ times the number of k-walks of the form Equation (1). ◀

We have shown in Lemma 5 that in order to bound the number of capped-k walks in G, it suffices to bound the number of k-walks where each vertex of the k-walk is required to belong to a specific "layer" set, where consecutive layers satisfy a degree regularity property. This motivates studying how large an edge density we can have between two such consecutive layers while avoiding C_{2k}'s. First, we need a minor modification of Theorem 1.

▶ **Corollary 6.** *Let G be a C_{2k}-free graph. Fix $A, B \subseteq V(G)$ not necessarily disjoint. Then,*

$$e(A, B) \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)}).$$

Proof. A well known fact (see e.g., [20]) in graph theory is that every m-edge graph has a bipartite subgraph with at least $m/2$ edges. Here we can apply the same idea. Let $C = A \cap B, A' = A \setminus C, B' = B \setminus C$. Form a partition of G as follows. Initialize $L = A', R = B'$

and then for every $c \in C$ place c in L with probability $1/2$ and in R otherwise. Then remove all edges not in $L \times R$. In expectation, at least half of the edges in $A \times B$ are preserved. Hence some setting of the random bits gives a bipartite subgraph with at least half the edges from $A \times B$ on which we can apply Theorem 1 to upper-bound $e(A,B)/2$ by $O(|L|+|R|+(|L||R|)^{(k+1)/(2k)})$ which is the desired quantity as $|L| \leq |A|$ and $|R| \leq |B|$. ◀

Now we prove a lemma to explain how consecutive layers in the decomposition of Lemma 5 interact.

▶ **Lemma 7.** *Let G be a C_{2k}-free graph with $\Delta(G) \leq m^{2/(k+1)}$. Fix $A \subseteq V(G)$ and $d \leq \Delta(G)$. Let B denote the set of vertices $v \in V(G)$ with $|N_A(v)| \in [d/2, d]$. Then,*

$$d\sqrt{|B|} \leq O(m^{\frac{1}{k+1}}\sqrt{|A|}).$$

Proof. Define $a = |A|, b = |B|, \Delta = \Delta(G)$. In this proof we will write $x \lesssim y$ to denote $x \leq O(y)$. We aim to show $d^2b/a \lesssim m^{2/(k+1)}$. This will follow from a simple chain of inequalities, with the main combinatorial ingredient being Corollary 6. In the inequalities we will use $\mathbb{1}_{[P]}$ to denote the indicator of whether P is true (i.e., it evaluates to 1 or 0 based on the truth of proposition P).

We have:

$$\begin{aligned}
d^2b/a &= \frac{(db)^2}{ab} \\
&\lesssim \min\left(\frac{d^2b}{a}, \frac{e(A,B)^2}{ab}\right) && B \text{ to } A \text{ degree regularity} \\
&\lesssim \min\left(\frac{d^2b}{a}, \frac{a^2+b^2+(ab)^{1+1/k}}{ab}\right) && \text{Corollary 6} \\
&\lesssim \min\left(\frac{d^2b}{a}, \frac{a}{b}\right) + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \text{re-arrange terms} \\
&\lesssim \sqrt{\frac{d^2b}{a}\cdot\frac{a}{b}} + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \min(x,y) \leq \sqrt{xy} \\
&\lesssim d + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \text{simplify} \\
&\lesssim \Delta + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && d \leq \Delta \\
&\lesssim \Delta + \frac{\Delta\cdot a}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && b \leq \left|\bigcup_{v\in A} N(v)\right| \leq \Delta\cdot a \\
&\lesssim m^{2/(k+1)} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \Delta \leq m^{2/(k+1)} \\
&\lesssim m^{2/(k+1)} + \mathbb{1}_{[a>dm^{1-2/(k+1)}]}\cdot\frac{d^2b}{a} + \mathbb{1}_{[a\leq dm^{1-2/(k+1)}]}\cdot(ab)^{1/k} && \text{split min} \\
&\lesssim m^{2/(k+1)} + \frac{d^2b}{dm^{1-2/(k+1)}} + (dm^{1-2/(k+1)}b)^{1/k} && \text{monotonicity} \\
&\lesssim m^{2/(k+1)} + \frac{m}{m^{1-2/(k+1)}} + (m^{2-2/(k+1)})^{1/k} && db \leq O(m) \\
&\lesssim m^{2/(k+1)} && \text{simplify.}
\end{aligned}$$

◀

Now we combine Lemma 5 and Lemma 7 to obtain Theorem 4.

Proof of Theorem 4. We apply Lemma 5 to obtain $V_*, d_*, G', X_1, \ldots, X_k \subseteq V(G'), d_1, \ldots, d_k$ with the properties described in Lemma 5. We can easily count the number of k-walks in $X_1 \times \cdots \times X_k \times V(G')$ due to the regularity property that consecutive X_i's enjoy; it is at most

$$|X_1| \cdot \prod_{j=1}^{k} d_j. \tag{2}$$

We now bound Equation (2) using Lemma 7, which implies that for each $i \in [k-1]$,

$$d_i \sqrt{|X_i|} \leq O(m^{1/(k+1)} \sqrt{|X_{i+1}|}). \tag{3}$$

Repeatedly applying Equation (3) to Equation (2) gives:

$$|X_1| \prod_{j=1}^{k} d_j \leq \sqrt{|X_1||X_k|} d_k \cdot O(m^{(k-1)/(k+1)}). \tag{4}$$

To bound Equation (4) we make two observations. First, note that $|X_k| d_k \leq O(m)$ because all $v \in X_k$ have degree at least $d_k/2$. Second, recall that $X_1 \subseteq V_*$, and $d_* \geq d_k$. Thus, $|X_1| d_k \leq |X_1| d_* \leq O(m)$. Applying this in Equation (4) gives

$$|X_1| \prod_{j=1}^{k} d_j \leq O(m^{2k/(k+1)}). \tag{5}$$

By Lemma 5, the number of capped k-walks in G is at most $\widetilde{O}(1)$ times Equation (5); this is the desired bound. ◀

[8] showed that Theorem 4 implies a C_{2k} detection algorithm. The C_{2k} detection algorithm is based on the following slightly more general lemma; this lemma is also the basis of our listing algorithm.

▶ **Lemma 8.** *Fix parameter $\Delta \in \mathbb{N}$, and graph G. Let W be the number of capped k-walks in G that start at a vertex with degree at most Δ. There is a (randomized) algorithm to list the C_{2k}'s in G in time $\widetilde{O}(m^2/\Delta + W + t)$.*

Proof. Label the vertices $v_1, \ldots, v_n$, where $v_i \succ v_j$ iff $i > j$. For each $i \in [n]$, define a subgraph G_i on vertices $\{v_1, \ldots, v_i\}$, consisting of the edges that are part of the k-step BFS out of vertex v_i. For $i = 1, 2, \ldots, n$, we will list the C_{2k}'s in G_i that involve v_i. Note that this will result in listing all C_{2k}'s; in particular, if v_i is the largest (by $\succ$) vertex in some C_{2k}, then we will list this C_{2k} on iteration i of our procedure. Let t_i denote the number of C_{2k}'s in G_i that involve v_i; we now discuss how to list these C_{2k}'s. To list the C_{2k}'s in G_i involving v_i, we use color coding. We color each vertex with a random color from $[2k]$. Then we restrict our search to **colorful** C_{2k}'s: C_{2k}'s with a vertex of every color. Listing the colorful C_{2k}'s involving v_i can be done in time $O(|E(G_i)| + t_i)$. Each C_{2k} becomes colorful with probability $2^{-2k} \geq \Omega(1)$, so by repeating this $\widetilde{O}(1)$ times, we will list all the C_{2k}'s in G_i involving v_i, with high probability.

Now we analyze the cost of our algorithm. Note that $|E(G_i)|$ is at most the number of capped k-walks that start at v_i. Let T be the largest i such that $\deg(v_i) \leq \Delta$. Then, by definition

$$\sum_{i=1}^{T} |E(G_i)| \leq O(W).$$

There are at most m/Δ vertices of degree larger than Δ. Thus,

$$\sum_{i=T+1}^{n} |E(G_i)| \leq O(m^2/\Delta).$$

Finally, we have $\sum_i t_i = O(t)$. Combined, we see that the algorithm runs in time $\widetilde{O}(t + W + m^2/\Delta)$. ◀

▶ **Corollary 9.** *There is a (randomized) $\widetilde{O}(m^{2k/(k+1)})$ time algorithm for C_{2k} detection.*

Proof. This follows by applying Lemma 8 with $\Delta = m^{2/(k+1)}$ (and terminating the algorithm if it runs for too long). ◀

Lemma 8 and Corollary 9 can be derandomized using standard techniques (perfect hash families etc) [4].

3 Progress Towards the Supersaturation Conjecture

Jin and Zhou observed ([13]) that the unbalanced supersaturation conjecture implies sparse listing algorithms for even cycles of any length; we give a simple proof of this in Section A. In this section we present partial progress towards proving the unbalanced supersaturation conjecture. We restrict our attention to **hexagons** (C_6's) because this is the smallest case where sparse listing algorithms are not known. In Section 4 we will show that our partial progress towards the supersaturation conjecture can be translated into partial progress towards listing hexagons in time $O(m^{1.5} + t)$. Before stating our result, we state some simple folklore facts about path supersaturation (see Section B for proofs).

▶ **Fact 10.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 2R$, then the number of 2-paths in $A \times B \times A$ is at least $L(m/L)(m/R)/2$.

▶ **Fact 11.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 50(R+L)$, then the number of 4-paths in $A \times B \times A \times B \times A$ is at least $\Omega(L(m/L)^2(m/R)^2)$.

Our partial supersaturation result for hexagons is as follows:

▶ **Theorem 12.** *Let G be a bipartite graph with vertex bipartition $A \sqcup B$ and t hexagons. Let $L = |A|, R = |B|$ with $100 \leq L \leq R$, and let $d_L = m/L, d_R = m/R$. Suppose $m/\lceil \log(L+R) \rceil^{10} > L + R + (LR)^{2/3}$. Then, there is an absolute constant $c > 0$ such that*

$$t \geq d_R^3 d_L^3 \cdot \min(1, d_R^3/L, d_L^2 d_R^2/L^2) \cdot c/\log^{70} n.$$

Proof. Define $n = 2^{\lceil \log(L+R) \rceil}$. The constraint $100(\log n)^{10} \leq m \leq n^2$ implies $n \geq 10^8$.

Organizing the Graph. Our approach is to first organize the graph and then find hexagons via two and three step BFS's. The start of our organization of the graph is the following claim:

▷ Claim 13. There exist disjoint sets $A', A'' \subseteq A$ and $B', B'' \subseteq B$, of sizes $L' = |A'|, L'' = |A''|, R' = |B'|, R'' = |B''|$, and there exist values d'_L, d''_L, d'_R such that:

- Vertices $v \in A'$ have $|N_{B'}(v)| \in [d'_L/4, d'_L]$ and $e(A', B') \geq m/\log^4 n$.
- Vertices $v \in B'$ have $|N_{A''}(v)| \in [d'_R/4, d'_R]$ and $e(B', A'') \geq m/\log^4 n$.
- Vertices $v \in A''$ have $|N_{B''}(v)| \in [d''_L/4, d''_L]$ and $e(A'', B'') \geq m/\log^4 n$.

Proof. Define $Y_1 = Y_3 = A, Y_2 = Y_4 = B$. Form vertex subsets X_1, X_2, X_3, X_4 as follows. Define $X_4 = Y_4$. For $i = 3, 2, 1$ define X_i as follows: Partition $y \in Y_i$ based on which dyadic interval $|N_{X_{i+1}}(y)|$ falls in. Let X_i be a part in this partition with the maximum number of edges to X_{i+1}. The number of parts in our dyadic partition (disregarding the $\{0\}$ part, which does not contribute edges) is $\log n$. Thus,

$$e(X_4, X_3) \geq m/\log n, e(X_3, X_2) \geq m/\log^2 n, e(X_2, X_1) \geq m/\log^3 n. \tag{6}$$

Now, assign each vertex $v \in V(G)$ a uniformly random color from $\{0, 1\}$. For $i = 1, 2$ define X'_i to be the subset of X_i colored 0. For $i = 3, 4$ define X'_i to be the subset of X_i colored 1. Note that the sets X'_1, X'_2, X'_3, X'_4 are disjoint by construction: in particular, even though $X_2 \subseteq X_4$, by the coloring $X'_2 \cap X'_4 = \emptyset$.

We now argue that there exists some coloring such that the sets $\{X'_i \mid i \in [4]\}$ have similar sizes and degree regularity properties to the sets $\{X_i \mid i \in [4]\}$. First, because for each i, $|X_i| \geq 100$, with probability at least .9999 we have $|X'_i| \geq |X_i|/4$ for each i. Now, fix $i \in [3], v \in X_i$. The value $|N_{X'_{i+1}}(v)|$ is is a Bernoulli random variable with $|N_{X_{i+1}}(v)|$ trials, and $p = 1/2$ success probability for each trial. By a Chernoff bound,

$$\Pr[|N_{X'_{i+1}}(v)| \leq |N_{X_{i+1}}(v)|/4] \leq \exp(-|N_{X_{i+1}}(v)|/16). \tag{7}$$

Now, we will argue that $|N_{X_{i+1}}(v)|$ is large enough that we can afford to union bound across Equation (7) for all $i \in [3], v \in X_i$. For $i = 1, 2, 3$ choose d_i such that each vertex $v \in X_i$ has $|N_{X_{i+1}}(v)| \in [d_i/2, d_i]$; the sets X_i were constructed so that this is possible. Now, Equation (6) implies for $i \in [3]$ that

$$d_i \geq \frac{m}{|X_i| \log^3 n} \geq \log^7 n. \tag{8}$$

Therefore, the probability that Equation (7) fails for any $i \in [3], v \in X_i$ is at most

$$3n \exp(-(\log^7 n)/32) \leq .001.$$

Thus with probability at least .99, for each $i \in [4]$, $|X'_i| \geq |X_i|/4$, and if $i \neq 4$ then each vertex $v \in X'_i$ has $|N_{X'_{i+1}}(v)| \in [d_i/4, d_i]$, and finally (using Equation (6))

$$|X'_i| d_i \geq e(X'_i, X'_{i+1}) \geq \frac{1}{16} e(X_i, X_{i+1}) \geq m/\log^4 n.$$

Thus, by the probabilistic method there is some coloring that gives these properties. Fix this coloring, and let $A' = X'_1, A'' = X'_3, B' = X'_2, B'' = X'_4$. These are precisely the sets we wanted to show exist. ◁

We continue to use the sets A', B', A'', B'' introduced in Claim 13 throughout the proof, along with the values L', R', L'', R'' and d'_L, d''_L, d'_R. A useful immediate corollary of Claim 13 is

▶ **Corollary 14.** *$d'_L \geq (L/L')d_L/\log^4 n \geq d_L/\log^4 n$, $d'_R \geq (R/R')d_R/\log^4 n \geq d_R/\log^4 n$, and*
$d''_L \geq (L/L'')d_L/\log^4 n \geq d_L/\log^4 n$.

Now we further organize the vertices. In what follows, we will partition vertices based on the **approximate value** of some quantity associated with the vertex, by which we mean which of the dyadic intervals $\mathcal{I}_n$ the quantity lies in. For vertex v and vertex subset X, define $N^2_X(v)$ to be the set of vertices $w \in X$ that are distance 2 from v.

Fix vertex $v \in A'$. Define (X_v, x_v, δ_v) (visualized in Figure 2) as follows: Partition $w \in N^2_{A''}(v)$ into **buckets** based on the approximate value of $|N_{B'}(v) \cap N_{B'}(w)|$. Let $X_v \subseteq A''$ be a bucket which maximizes the number of edges between X_v and $N_{B'}(v)$. Let $x_v = |X_v|$, and choose $\delta_v \in \{1\} \cup \{4, 8, 16 \ldots, n\}$ such that each $w \in X_v$ has $|N_{B'}(v)| \in [\delta_v/2, \delta_v]$ (this is possible by the definition of $\mathcal{I}_n$, and the fact that the $\{0\}$ bucket does not contribute any edges). Then,

$$\delta_v x_v \geq e(X_v, N_{B'}(v)) \geq \frac{1}{\log n} e(N_{B'}(v), A'') \geq \frac{1}{16 \log n} \cdot d'_L d'_R. \tag{9}$$

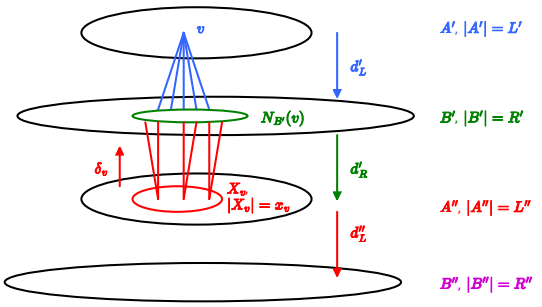


Figure 2 Organizing the graph.

Next, we assign each vertex $r \in N_{B'}(v)$ a tuple $(Y_{rv}, y_{rv}, \lambda_{rv})$ (visualized in Figure 3) as follows: Partition $z \in N^2_{B''}(r)$ into buckets based on the approximate value of $|N_{X_v}(r) \cap N_{X_v}(z)|$. Let $Y_{rv} \subseteq B''$ be a bucket which maximizes the number of edges between Y_{rv} and $N_{X_v}(r)$. Let $y_{rv} = |Y_{rv}|$ and choose $\lambda_{rv} \in \{1\} \cup \{4, 8, 16 \ldots, n\}$ such that each $z \in B''$ has $|N_{X_v}(r) \cap N_{X_v}(z)| \in [\lambda_{rv}/2, \lambda_{rv}]$. Then,

$$\lambda_{rv} y_{rv} \geq e(N_{X_v}(r), Y_{rv}) \geq \frac{1}{\log n} e(N_{X_v}(r), B''). \tag{10}$$

Finally, partition $r \in N_{B'}(v)$ into buckets based on the approximate values of $y_{rv}, \lambda_{rv}, |N_{X_v}(r)|$. Let $B'_v \subseteq B'$ be a bucket which maximizes the number of edges between B'_v and X_v. Choose y_v, λ_v, β_v (visualized in Figure 3) such that each $r \in B'_v$ has $y_{rv} \in [y_v/2, y_v]$, $\lambda_{rv} \in [\lambda_v/2, \lambda_v]$, and $|N_{X_v}(r)| \in [\beta_v/2, \beta_v]$. Then,

$$e(B'_v, X_v) \geq \frac{1}{\log^3 n} e(N_{B'}(v), X_v). \tag{11}$$

Counting Hexagons in the Organized Graph. Let $\mathsf{hex}(G)$ denote the number of hexagons in G. With the graph nicely organized we are prepared to bound $\mathsf{hex}(G)$. For vertex $v \in A'$, let $\mathsf{hex}(v)$ denote the number of hexagons in G where v is the only A' vertex participating in the hexagon. One natural way of bounding $\mathsf{hex}(G)$ is by $\mathsf{hex}(G) \geq \sum_{v \in A'} \mathsf{hex}(v)$. We will split into cases based on the characteristics of the 3-step BFS out of each vertex v and attempt to show that $\mathsf{hex}(v)$ is large. However, in one of the cases we will show $\mathsf{hex}(G)$ is large directly from the 3-step BFS out of a single vertex, rather than bounding $\mathsf{hex}(v)$. We now begin to consider 3-step BFS's based on their characteristics. See Figure 4 and Figure 5 for an illustration of our approach.

▷ Claim 15. Suppose vertex $v \in A'$ has $x_v d''_L/4 > 2R$ and $\delta_v \neq 1$. Then $\mathsf{hex}(v) \geq \Omega(d^3_R d^3_L/L')/\log^{26} n$.

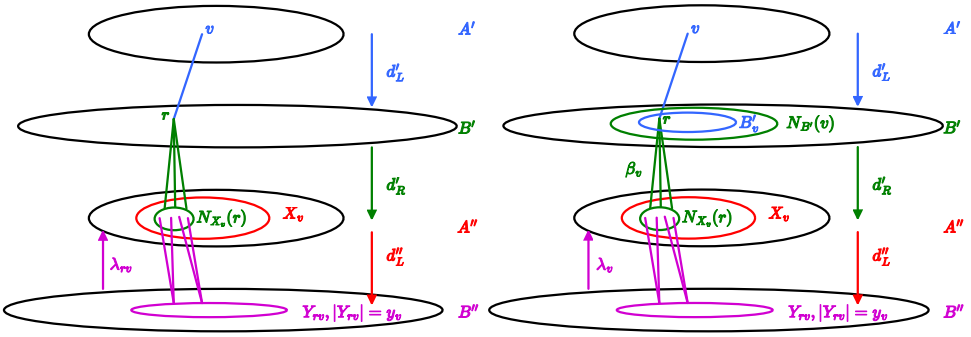


Figure 3 Organizing the graph.

Proof. Recall that $e(X_v, B'') \geq x_v d''_L/4$, which by assumption of the claim is at least $2R$. Thus, Fact 10 implies that the number of 2-paths in $X_v \times B'' \times X_v$ is at least

$$\Omega((x_v d''_L)^2/R). \tag{12}$$

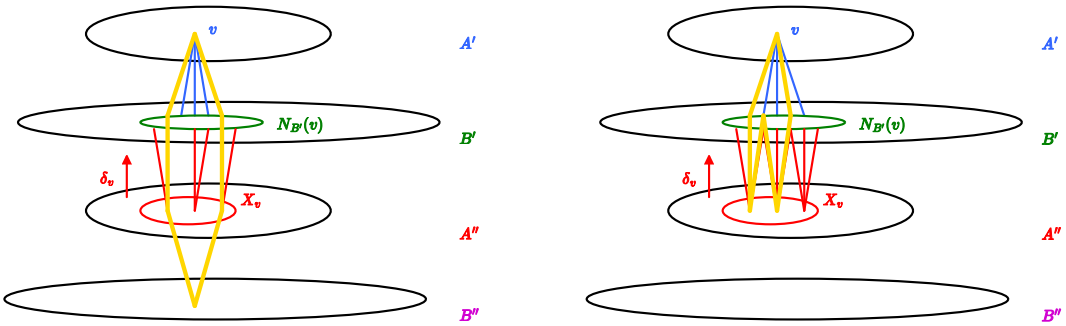
Given a two path $(w_1, z, w_2) \in X_v \times B'' \times X_v$, there are at least $(\delta_v/2)(\delta_v/2-1)$ ways to choose distinct $r_1 \in N_{B'}(w_1), r_2 \in N_{B'}(w_2)$. In this claim we are assuming $\delta_v \neq 1$ and consequentially $\delta_v \geq 4$. Thus, $(\delta_v/2)(\delta_v/2-1) \geq \delta_v^2/8$. Choosing r_1, r_2 in this manner, the vertices $(v, r_1, w_1, z, w_2, r_2)$ form a hexagon containing a single vertex from A'. Combined with Equation (12) this implies

$$\mathsf{hex}(v) \geq \Omega(\delta_v^2 (x_v d''_L)^2/R). \tag{13}$$

Using Equation (9) and Corollary 14 in Equation (13) gives

$$\mathsf{hex}(v) \geq \frac{1}{\log^2 n}\Omega((d'_L d'_R d''_L)^2/R) \geq \frac{1}{\log^{26} n}\Omega(d_R^3 d_L^3/L').$$

◁



(a) Claim 15. **(b)** Claim 19.

Figure 4 Two methods for showing that $\mathsf{hex}(v)$ is large.

Before proceeding to the next case, we make some simple observations.

▶ **Fact 16.** $d_R, d_L \geq \log^{10} n \geq 10^{14}$.

▶ **Corollary 17.** $d'_L d'_R d''_L/R \geq 10^{15} \log n$.

Proof. By Corollary 14 and Fact 16 we have

$$d'_L d''_L d'_R/R \geq (d_L^2 d_R/R)/\log^{12} n \geq (m^3/(LR)^2)/\log^{12} n \geq \log^{18} n \geq 10^{15}\log n. \quad \blacktriangleleft$$

▶ **Corollary 18.** $d'_R \geq 10^7 \log n$.

Proof. By Corollary 14 and Fact 16 we have

$$d'_R \geq d_R/\log^4 n \geq \log^6 n \geq 10^7 \log n. \quad \blacktriangleleft$$

▷ Claim 19. Suppose vertex $v \in A'$ has $x_v d''_L/4 \leq 2R$. Then $\mathsf{hex}(v) \geq \Omega((d_L d_R)^3 d_R^3/(LL'))/\log^{40} n$.

Proof. We claim that if x_v is this small then we have supersaturation of 4-paths between $N_{B'}(v)$ and X_v that start in $N_{B'}(v)$. Indeed, Equation (9), Corollary 17, Corollary 18 and our assumption that x_v is small imply

$$e(N_{B'}(v), X_v) \geq \frac{1}{16\log n} d'_L d'_R \geq 10^{12} R/d''_L + 10^5 d'_L \geq 50(d'_L + 8R/d''_L) \geq 50(d'_L + x_v).$$

Applying Fact 11, Equation (9), the hypothesis that x_v is small, and Corollary 14, the number of such 4-paths is at least

$$\frac{1}{\log^4 n}\Omega\left(\frac{(d'_L d'_R)^4}{d'_L x_v^2}\right) \geq \Omega((d_L d_R)^3 d_R^3/(LL'))/\log^{40} n.$$

Each of these 4-paths paired with v gives a hexagon that contributes to $\mathsf{hex}(v)$. This yields the desired bound on $\mathsf{hex}(v)$. ◁

The remaining two cases for what the BFS out of v looks like will both have $\delta_v = 1$. When $\delta_v = 1$, we have the following interesting property:

▷ Claim 20. Fix $v \in A'$ with $\delta_v = 1$. Then, $|B'_v| \geq d'_L/\log^5 n$ and $\beta_v \geq d'_R/\log^5 n$.

Proof. We count $e(B_{v'}, X_v)$ in two ways. First,

$$e(B_{v'}, X_v) \leq |B_{v'}|\beta_v. \tag{14}$$

Second, combining Equation (11) and Equation (9) we have

$$e(B_{v'}, X_v) \geq \frac{1}{\log^3 n} x_v \geq \frac{1}{16\log^4 n} d'_L d'_R. \tag{15}$$

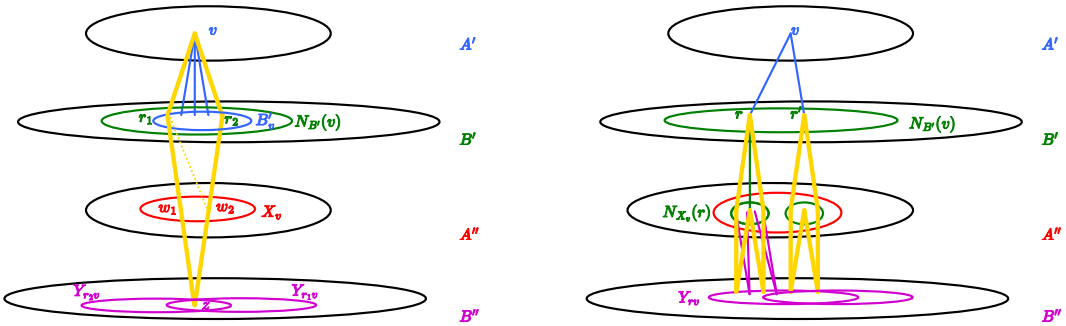
Combining Equation (14) and Equation (15) gives

$$|B'_v|\beta_v \geq \frac{1}{\log^5 n} d'_L d'_R. \tag{16}$$

Now we recall upper bounds for $|B_{v'}|, \beta_v$. Applying $\beta_v \leq d'_R$ in Equation (16) gives $|B'_v| \geq d'_L/\log^5 n$. Applying $|B'_v| \leq d'_L$ in Equation (16) gives $\beta_v \geq d'_R/\log^5 n$. ◁

We are now ready to handle the cases where $\delta_v = 1$.

▷ Claim 21. Fix $v \in A'$ and suppose $y_v d'_L > (4\log^5 n)R$ and $\delta_v = 1$. Then $\mathsf{hex}(v) \geq \Omega((d_L^3 d_R^3)/L')/\log^{46} n$.



(a) Claim 21. **(b)** Claim 22.

Figure 5 Handling $\delta_v = 1$.

Proof. We construct an auxiliary graph H between B'_v and B'' as follows: place an edge between $r \in B'_v$ and $z \in B''$ if r, z have a common neighbor in X_v. We have, $|E(H)| \geq |B'_v| y_v/2$. By Claim 20, and the hypothesis of the claim we have

$$|E(H)| \geq \frac{1}{2\log^5 n} d'_L y_v \geq 2R. \tag{17}$$

Thus, applying Fact 10 to Equation (17), the number of 2-paths in H starting in B'_v, is at least

$$\frac{1}{\log^{10} n} \Omega\left((d'_L y_v)^2/R.\right). \tag{18}$$

Now, fix some 2-path $(r_1, z, r_2) \in B'_v \times B'' \times B'_v$ in H. There are at least $\lceil \lambda_v/4 \rceil^2$ pairs $w_1, w_2 \in X_v$ such that $w_i \in N(z) \cap N(r_i)$ for $i = 1, 2$. We claim that for each such w_1, w_2, the tuple $(v, r_1, w_1, z, w_2, r_2)$ is a hexagon. This is not immediately obvious: we need to rule out the possibility of $w_1 = w_2$ (in which case we would have a 4-cycle with a dangling edge rather than a true hexagon). Fortunately, we can rule this out: the fact $\delta_v = 1$ precisely means that for $i = 1, 2$, w_i has only 1 neighbor in B'_v. Thus, we get at least $\lceil \lambda_v/4 \rceil^2$ times the quantity in Equation (18) many hexagons. That is,

$$\mathsf{hex}(v) \geq \frac{1}{\log^{10} n} \Omega((d'_L y_v \lambda_v)^2/R). \tag{19}$$

Now, recall that Equation (10) gives a bound on $y_v \lambda_v$: it asserts that for any $r \in B'_v$ we have

$$y_v \lambda_v \geq y_{rv} \lambda_{rv} \geq \frac{1}{\log n} e(N_{X_v}(r), B'') \geq \frac{1}{\log n} |N_{X_v}(r)| d''_L/4 \geq \frac{1}{\log n} (\beta_v/2) d''_L/4. \tag{20}$$

Now, using Claim 20 we have

$$y_v \lambda_v \geq \frac{1}{\log^6 n} \Omega(d'_R d''_L). \tag{21}$$

Now, we can bound Equation (19) by

$$\mathsf{hex}(v) \geq \frac{1}{\log^{22} n} \Omega((d'_L d'_R d''_L)^2)/R.$$

Now, using Corollary 14 we have

$$\begin{aligned}\mathsf{hex}(v) &\geq \frac{1}{\log^{22} n}\Omega((d'_L d'_R d''_L)^2)/R \\ &\geq \frac{1}{\log^{26} n}\Omega(d'_L(d'_R d''_L)^2(m/L')/R) \\ &\geq \frac{1}{\log^{46} n}\Omega(d_L(d_R d_L)^2(m/L')/R) \\ &= \frac{1}{\log^{46} n}\Omega(d_L^3 d_R^3/L').\end{aligned}$$

◁

The final case is slightly different. Instead of bounding $\mathsf{hex}(v)$ we will directly bound $\mathsf{hex}(G)$.

▷ Claim 22. Fix $v \in A'$ with $\delta_v = 1$, $y_v d'_L \leq (4\log^5 n)R$, and $x_v d''_L > 8R$. Then $\mathsf{hex}(G) \geq \Omega((d_L d_R)^5/L^2)/\log^{67} n$.

Proof. We will find hexagons of the following form: Take $r \in B'_v$, and then let $(x_1, x_2, x_3, x_4, x_5)$ be a 4-path between $N_{X_v}(r)$ and Y_{rv} (starting in $N_{X_v}(r)$). Then, $(r, x_1, x_2, x_3, x_4, x_5)$ is a hexagon. We claim that we have 4-path supersaturation between $N_{X_v}(r)$ and Y_{rv}. Indeed, by Equation (10) we have

$$e(N_{X_v}(r), Y_{rv}) \geq \frac{1}{\log n} e(N_{X_v}(r), B'') \geq \frac{1}{4\log n}|N_{X_v}(r)|d''_L \geq \frac{1}{8\log n}\beta_v d''_L.$$

Then, by Claim 20 we have

$$e(N_{X_v}(r), Y_{rv}) \geq \frac{1}{8\log^6 n} d'_R d''_L. \tag{22}$$

Now, recall that $|N_{X_v}(r)| \leq d'_R$, so Equation (22) together with Corollary 14 and Fact 16 imply $e(N_{X_v}, Y_{rv}) \geq 100|N_{X_v}(r)|$. Next, observe that by Corollary 14, Equation (22) implies

$$|Y_{rv}| \leq y_v \leq (4\log^5 n)R/d'_L \leq \frac{1}{4\log n}\frac{LR}{m} \leq \frac{1}{4\log^{31} n}\frac{m^2}{LR} \leq \frac{d_L d_R}{\log^{31} n} \leq \frac{d'_R d''_L}{800\log^6 n} \leq \frac{e(N_{X_v}(r), Y_{rv})}{100}.$$

Combined, we have

$$e(N_{X_v}(r), Y_{rv}) \geq 50(|Y_{rv}| + |N_{X_v}(r)|),$$

which is the required condition for 4-path supersaturation. Now, applying Fact 11 to Equation (22) the number of 4-paths that we get is:

$$\Omega\left(\frac{e(N_{X_v}(r), Y_{rv})^4}{|N_{X_v}(r)||Y_{rv}|^2}\right) \geq \frac{1}{\log^{24} n}\Omega\left(\frac{(d'_R d''_L)^4}{\beta_v y_v^2}\right) \tag{23}$$

Applying our assumption on y_v and the bound $\beta_v \leq d'_R$ gives

$$\frac{1}{\log^{34} n}\Omega\left(\frac{(d'_R d''_L)^4}{d'_R(R/d'_L)^2}\right) \tag{24}$$

Applying Corollary 14 to Equation (24), the number of such 4-paths is at least

$$\frac{m^9}{L^6 R^5}\Omega(1/\log^{61} n).$$

Each $r \in B'_v$ contributes this many hexagons, and the contributed hexagons are distinct. Applying Claim 20 we have

$$\mathsf{hex}(G) \geq |B'_v|\frac{m^9}{L^6 R^5}\Omega(1/\log^{61} n) \geq \frac{m^{10}}{L^7 R^5}\Omega(1/\log^{67} n) \geq (d_R d_L)^3(d_L d_R/L)^2/\log^{67} n.$$ ◁

Now, we can easily combine Claim 15, Claim 19, Claim 21, Claim 22 to deduce the theorem. Indeed, if any vertex $v \in A'$ falls into the case described by Claim 22 then have the desired bound on $\mathsf{hex}(G)$. Otherwise, each vertex must fall into one of the cases covered in Claim 15, Claim 19, Claim 21. Picking whichever case is most common of these, we can sum over the $\mathsf{hex}(v)$ bounds for at least $|A'|/3 = L'/3$ many $v \in A'$. This gives the desired bound on $\mathsf{hex}(G)$. ◀

4 Progress Towards Listing Algorithms

In this section we describe how our partial progress towards Conjecture 25 in Theorem 12 can be used together with our simplified analysis capped k-walks analysis from Section 2 to give improved listing algorithms for C_6's in sparse graphs. Our main result is:

▶ **Theorem 23.** *There is an algorithm for listing C_6's in time $\widetilde{O}(m^{1.6} + t)$.*

Theorem 23 follows immediately from Lemma 8 with $\Delta = m^{.4}$ and the following lemma:

▶ **Lemma 24.** *Fix G with $\Delta(G) \leq m^{.4}$. The number of capped 3-walks in G is at most $\widetilde{O}(m^{1.6} + t)$.*

Proof. If $n \leq 1000$ the result is trivial; we assume this is not the case. Fix $\lambda = 200\lceil \log n \rceil^{15}$. As in Theorem 4 and Theorem 28 we use Lemma 5 to obtain $\mathcal{B} = (V_*, d_*, G', X_1, X_2, X_3, d_1, d_2, d_3)$ with the properties described in Lemma 5. The number of 3-walks in $X_1 \times X_2 \times X_3 \times V(G')$ is at most $|X_1| \cdot d_1 d_2 d_3$; recall that the number of capped 3-walks in G is at most $\text{polylog}(n)|X_1|d_1d_2d_3$ – thus, in the rest of this proof we'll focus on bounding $|X_1|d_1d_2d_3$. At this point we break into 576 cases (with the help of a computer). A **case** is parameterized by a tuple $(B_{12}, B_{23}, D_{12}, D_{23}, U_{12}, U_{23}, A_{12}, A_{23})$, whose entries can take on the following values:

- $B_{12} \in \{1, 2\}, B_{23} \in \{2, 3\}$. B_{12} indicates which of X_1, X_2 is bigger (B stands for "bigger"). More precisely, $|X_{B_{12}}| = \max(|X_1|, |X_2|)$. B_{23} is defined analogously.
- $D_{12}, D_{23} \in \{1, 2, 3\}$. D_{12}, D_{23} indicate the "density regime" (D stands for "density") that the graphs between (X_1, X_2) and (X_2, X_3) fall into. This means, letting $R = \max(|X_1|, |X_2|), L = \min(|X_1|, |X_2|), m_{12} = e(X_1, X_2)$ and letting $v_1 = 1, v_2 = (d_R^3/L)/\lambda^3, v_3 = (d_L^2 d_R^2/L^2)/\lambda^4$, we have that $v_{D_{12}} = \min(v_1, v_2, v_3)$. D_{23} is defined analogously.
- $U_{12} \in \{0, 1\}, U_{23} \in \{0, 1\}$. U_{12} indicates whether the graph between X_1, X_2 is "balanced" or "unbalanced" (U stands for "unbalanced"). More precisely, U_{12} is 1 if $R \geq (RL)^{2/3}$ and 0 if $R \leq (RL)^{2/3}$. U_{23} is defined analogously.
- $A_{12}, A_{23} \in \{0, 1\}$. A_{12} indicates whether we can charge any hexagons to the graph between X_1, X_2 (A stands for "any"). More precisely, A_{12} is 1 if $m \geq \lambda \max(|X_1|, |X_2|, (|X_1||X_2|)^{2/3})$ and 0 if $m \leq \lambda \max(|X_1|, |X_2|, (|X_1||X_2|)^{2/3})$. A_{23} is defined analogously.

Note that $\mathcal{B}$ falls into at least one case (and the cases are slightly overlapping on the equality cases). Now, let $\mathcal{C} = (B_{12}, B_{23}, D_{12}, D_{23}, U_{12}, U_{23}, A_{12}, A_{23}) \in \{1, 2\} \times \{2, 3\} \times \{1, 2, 3\}^2 \times \{0, 1\}^4$ be a case that describes $\mathcal{B}$. For $i = 1, 2, 3$, let $x_i = \log_m(|X_i|), \delta_i = \log_m(d_i/\lambda)$. Let $\delta_* = \log_m(d_*/\lambda)$, and let τ denote $\log_m(t/c)$ where t is the number of C_6's in G, and c is the constant from Theorem 12. We now outline a series of *linear* constraints that $(x_1, x_2, x_3, \delta_1, \delta_2, \delta_3, \tau)$ must satisfy.

$$x_1, x_2, x_3 \in [0,1] \qquad \forall i, |X_i| \leq n \leq m.$$
$$\delta_*, \delta_1, \delta_2, \delta_3 \in [0, .4] \qquad \Delta(G) \leq m^{.4}.$$
$$\tau \geq 0$$
$$\forall i \in [3], \delta_i \leq \delta_*. \qquad \Delta(G') \leq \delta_*.$$
$$\delta_* + x_1 \leq 1, \qquad |X_1|\delta_* \leq 2m$$
$$\forall i \in [3], \delta_i + x_i \leq 1 \qquad \forall i, |X_i| d_i \leq 2m.$$

Next, based on B_{12}, B_{23} we get ordering relationships between x_1, x_2, x_3. For instance, if $B_{12} = 1, B_{23} = 3$ then we add the constraints

$$x_1 \geq x_2, x_2 \leq x_3.$$

We get more constraints from U_{12}, U_{23}. For instance, if $U_{12} = 0, U_{23} = 1$ we add the following constraints:

$$\max(x_1, x_2) \leq 2\min(x_1, x_2), \max(x_2, x_3) \geq 2\min(x_2, x_3).$$

Note that while these constraints appear non-linear (due to the $\max, \min$) they are actually linear, because B_{12}, B_{23} already specify orders between x_1, x_2, and between x_2, x_3! So, based on the value of B_{12}, B_{23} we could "unroll" these min / max's to the appropriate x_i; we decline to do so here for clarity.

Next, we implement constraints based on A_{12}, A_{23}. If $A_{12} = 1$ the constraint will be

$$x_1 + \delta_1 \geq \max(x_1, x_2, (2/3)(x_1 + x_2)).$$

The constraint would be flipped if $A_{12} = 0$. The constraint is analogous for A_{23}. Again, the max can be unrolled due to the value of the max being determined by B_{12}, U_{12}.

Next we implement constraints based on D_{12}, D_{23}. For instance, if $D_{12} = 1$ we would add constraints (where, as always, the $\max, \min$'s can be unrolled):

$$0 \leq 3(x_1+\delta_1) - 3\max(x_1, x_2) - \min(x_1, x_2), 0 \leq 4(x_1+\delta_1) - 2\max(x_1, x_2) - 4\min(x_1, x_2).$$

Collecting Hexagons. Now, with all these constraints in place, we will use Theorem 12 to obtain a lower bound on the number of hexagons in our graph. Then we'll set up an optimization problem that tells us the slowest possible time for our algorithm, observe that this optimization problem is an LP in each case, and solve the LPs on a computer to find a bound on the performance of our algorithm.

If $A_{12} = 1$, then letting $\ell = \min(x_1, x_2), r = \max(x_1, x_2), e = x_1 + \delta_1$ we add the constraint

$$\tau \geq 6e - 3(r + \ell) + \min(0, 3e - 3r - \ell, 4e - 4\ell - 2r).$$

This constraint is implied by Theorem 12, and we satisfy the pre-condition of Theorem 12 because $A_{12} = 1$. Note that as usual, the min / max's can be unrolled based on the case. If $A_{23} = 1$ we add an analogous constraint on τ.

Bounding capped-3 walks. Now, we want to answer the following question: "subject to $(x_1, x_2, x_3, \delta_*, \delta_1, \delta_2, \delta_3)$ satisfying the above constraints, is it possible that the number of 3-walks in $X_1 \times X_2 \times X_3 \times V(G')$ is larger than $\widetilde{O}(m^{1.6} + t)$? We can answer this question in

the negative by requiring the number of 3-walks to be more than $\Omega(t \log^{45} n)$, and showing that subject to this constraint, the number of 3-walks cannot be larger than $O(m^{1.6})$. In our linear programming formulation, this corresponds to adding the constraint

$$x_1 + \delta_1 + \delta_2 + \delta_3 \geq \tau,$$

and optimizing the following objective function, which is the number of 3-walks (up to $\operatorname{poly}\log n$ factors):

$$x_1 + \delta_1 + \delta_2 + \delta_3.$$

If the objective function optimizes to a number which is at most 1.6, then it will prove that the number of 3-walks of the specified form cannot be larger than $\widetilde{O}(m^{1.6} + t)$.

So, we can prove the lemma by iterating over the 576 possible cases, defining the appropriate linear program in each case, and checking that the optimum of the linear program is at most 1.6. We wrote a simple program [18] to iterate over the cases and solve the linear programs. Running the code we find that the optimum of each linear program is at most 1.6, as desired. We leave developing a human interpretable proof as an open question. ◀

References

1. Amir Abboud, Karl Bringmann, and Nick Fischer. Stronger 3-sum lower bounds for approximate distance oracles via additive combinatorics. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 391–404, 2023. doi:10.1145/3564246.3585240.
2. Amir Abboud, Karl Bringmann, Seri Khoury, and Or Zamir. Hardness of approximation in p via short cycle removal: cycle detection, distance oracles, and beyond. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1487–1500, 2022. doi:10.1145/3519935.3520066.
3. Amir Abboud, Seri Khoury, Oree Leibowitz, and Ron Safier. Listing 4-cycles. *arXiv preprint*, 2022. doi:10.48550/arXiv.2211.10022.
4. Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
5. Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.
6. John A Bondy and Miklós Simonovits. Cycles of even length in graphs. *Journal of Combinatorial Theory, Series B*, 16(2):97–105, 1974.
7. Karl Bringmann and Egor Gorbachev. A fine-grained classification of subquadratic patterns for subgraph listing and friends. *CoRR*, abs/2404.04369, 2024. doi:10.48550/arXiv.2404.04369.
8. Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Morten Stöckel. Finding even cycles faster via capped k-walks, 2017. arXiv:1703.10380.
9. Pierre-Louis Giscard, Paul Rochet, and Richard C Wilson. Evaluating balance on social networks from their simple cycles. *Journal of Complex Networks*, 5(5):750–775, 2017. doi:10.1093/COMNET/CNX005.
10. Tao Jiang and Liana Yepremyan. Supersaturation of even linear cycles in linear hypergraphs. *Combinatorics, Probability and Computing*, 29(5):698–721, 2020. doi:10.1017/S0963548320000206.
11. Ce Jin, Virginia Vassilevska Williams, and Renfei Zhou. Listing 6-cycles. In *Proceedings of the Symposium on Simplicity in Algorithms (SOSA 2024)*, pages 19–27, 2024. doi:10.1137/1.9781611977936.3.
12. Ce Jin and Yinzhan Xu. Removing additive structure in 3sum-based reductions. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 405–418, 2023. doi:10.1145/3564246.3585157.

13 Ce Jin and Renfei Zhou. Personal communication, 2024.

14 Steffen Klamt and Axel von Kamp. Computing paths and cycles in biological interaction graphs. *BMC bioinformatics*, 10:1–11, 2009. doi:10.1186/1471-2105-10-181.

15 Andrea Lincoln and Nikhil Vyas. Algorithms and Lower Bounds for Cycles and Walks: Small Space and Sparse Graphs. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2020.11.

16 V. Vassilevska. Efficient algorithms for path problems. *Ph.D. Thesis in Computer Science, Carnegie Mellon University*, 2008.

17 Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5), August 2018. doi:10.1145/3186893.

18 Alek Westover. Listing-c6s-lp. https://github.com/awestover/listing-C6s-LP/tree/main, 2024. Accessed: 2024-08-08.

19 Raphael Yuster and Uri Zwick. Finding even cycles even faster. *SIAM Journal on Discrete Mathematics*, 10(2):209–222, 1997. doi:10.1137/S0895480194274133.

20 Yufei Zhao. *Graph Theory and Additive Combinatorics: Exploring Structure and Randomness*. Cambridge University Press, 2023.

A Listing C_{2k}'s Using the Supersaturation Conjecture

Jin and Zhou [13] observed that the unbalanced supersaturation conjecture would allow one to obtain conditionally optimal sparse / dense listing algorithms for C_{2k}'s. In this section we give a simple proof of this observation, using the techniques of Section 2. Recall the unbalanced supersaturation conjecture:

▶ **Conjecture 25.** (The Unbalanced Supersaturation Conjecture [Jin and Zhou'24]) *Let G be an m edge bipartite graph with vertex bipartition $A \sqcup B$, with t C_{2k}'s. Suppose $m \geq 100k(|A| + |B| + (|A||B|)^{(k+1)/2k})$. Then,*

$$t \geq \Omega\left(\frac{m^{2k}}{|A|^k|B|^k}\right).$$

Conditional on Conjecture 25 we can follow the same proof strategy from Section 2 to obtain a C_{2k} listing algorithm. First, we need a simple consequence of Conjecture 25.

▶ **Corollary 26.** *Assume Conjecture 25. Fix G and $A, B \subseteq V(G)$* not necessarily disjoint. *Then,*

$$m \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)} + t^{1/(2k)}\sqrt{|A||B|}).$$

Proof. As in Corollary 6, we pass to a bipartite subgraph with at least $m/2$ edges, while only decreasing $|A|, |B|, t$. Then we bound $m/2$ using Conjecture 25. ◀

Next we prove an analogue of Lemma 7, for analyzing the structure between two layers in the decomposition of Lemma 5.

▶ **Lemma 27.** *Assume Conjecture 25 Let graph G have $\Delta(G) \leq m^{2/(k+1)}$. Fix $A \subseteq V(G)$ and $d \leq \Delta(G)$. Let B denote the set of vertices $v \in V(G)$ that have $|N_A(v)| \in [d/2, d]$. Then,*

$$d\sqrt{|B|}/\sqrt{|A|} \leq O(m^{\frac{1}{k+1}} + t^{1/(2k)}).$$

Proof. The proof is nearly identical to that of Lemma 7. The only difference is that instead of applying the bound

$$e(A,B) \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)}),$$

which applied when we assumed the graph was C_{2k}-free, we apply Corollary 26, which gives:

$$e(A,B) \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)} + t^{1/(2k)}\sqrt{|A||B|}).$$

Substituting this new bound on $e(A,B)$ into the proof of Lemma 7 and otherwise leaving the analysis unchanged gives the desired bound on $d\sqrt{|B|/|A|}$. ◀

Now we deduce the analog of Theorem 4.

▶ **Theorem 28.** *Assume Conjecture 25. Let G be a graph with $\Delta(G) \leq m^{2/(k+1)}$. Then, the number of capped k-walks in G is at most $\widetilde{O}(m^{2k/(k+1)} + t)$.*

Proof. Just like in the proof of Theorem 4 we apply Lemma 5 to obtain V_*, d_*, G', $X_1, \ldots, X_k \subseteq V(G'), d_1, \ldots, d_k$ with the properties described in Lemma 5. The number of k-walks in $X_1 \times \cdots \times X_k \times V(G')$ is at most $|X_1| \cdot \prod_{j=1}^{k} d_j$. In Theorem 4 we bounded this product by repeatedly applying Lemma 7 (and then using an additional observation to bound $\sqrt{|X_1||X_k|}d_k$). Using Lemma 27 in place of Lemma 7 we have:

$$|X_1| \prod_{j=1}^{k} d_j \leq O((m^{1/(k+1)} + t^{1/(2k)})^{k-1} \cdot m). \tag{25}$$

Applying Hölder's Inequality to Equation (25) proves the desired bound on capped k-walks. ◀

Finally, we have the analog of Corollary 9.

▶ **Corollary 29.** *Assume Conjecture 25. Then, there is an algorithm for listing C_{2k}'s with running time $\widetilde{O}(t + \min(m^{2k/(k+1)}, n^2))$.*

Proof. First, we note that it is sufficient to establish the existence of an $\widetilde{O}(t + m^{2k/(k+1)})$ time algorithm. This is because balanced supersaturation results (Theorem 2) imply that if $m^{2k/(k+1)} > \Omega(n^2)$, meaning $m > \Omega(n^{1+1/k})$, then $t > \Omega(m^{2k/(k+1)})$, and so the t term dominates the running time. Now we give a listing algorithm with running time $\widetilde{O}(t + m^{2k/(k+1)})$.

Set $\Delta = m^{2/(k+1)}$. Assuming Conjecture 25, Theorem 28 implies that there are at most $\widetilde{O}(m^{2k/(k+1)} + t)$ capped-k walks starting at a vertex of degree at most Δ. Then, applying Lemma 8 gives a listing algorithm with the desired running time. ◀

B Folklore Results about Path Supersaturation

▶ **Fact 30.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 2R$, then the number of 2-paths in $A \times B \times A$ is at least $L(m/L)(m/R)/2$.

Proof. Using Cauchy-Schwarz, the number of 2-paths is at least

$$\sum_{v \in B} \deg(v)(\deg(v) - 1) \geq \frac{1}{R}\left(\sum_{v \in B} \deg(v)\right)^2 - \sum_{v \in B} \deg(v) \geq m(m/R - 1) \geq m^2/(2R). \quad ◀$$

▶ **Fact 31.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 50(R+L)$, then the number of 4-paths in $A \times B \times A \times B \times A$ is at least $\Omega(L(m/L)^2(m/R)^2)$.

Proof. Throughout the proof we use the notation $\deg_S(v) = |N(v) \cap S|$.

- Let B_1 be the set of vertices $v \in B$ with $\deg_A(v) \geq e(A,B)/(2|B|)$.
- Let A_1 be the set of vertices $v \in A$ with $\deg_{B_1}(v) \geq e(A,B_1)/(2|A|)$.
- Let B_2 be the set of vertices $v \in B_1$ with $\deg_{A_1}(v) \geq e(A_1,B_1)/(2|B_1|)$.

Observe that $e(A,B_1) \geq e(A,B)/2, e(A_1,B_1) \geq e(A,B)/4, e(A_1,B_2) \geq e(A,B)/8$. We now demonstrate that there are a large number of 4-paths starting in A. We can generate such a 4-path as follows:

- Fix an edge $(v_5, v_4) \in A_1 \times B_2$.
- Take $v_3 \in N_{A_1}(v_4)$, $v_2 \in N_{B_1}(v_3), v_1 \in N_A(v_2)$.

The number of such walks is at least:

$$\frac{e(A,B)}{8}\left(\frac{e(A,B)}{8|B_1|}-1\right)\left(\frac{e(A,B)}{4|A|}-1\right)\left(\frac{e(A,B)}{2|B|}-2\right) \geq \Omega(m^4/(LR^2)).$$ ◀