Beating Treewidth for Average-Case Subgraph Isomorphism

Gregory Rosenthal
University of Toronto, Canada
http://www.cs.toronto.edu/~rosenthal/
rosenthal@cs.toronto.edu

Abstract

For any fixed graph G, the subgraph isomorphism problem asks whether an n-vertex input graph has a subgraph isomorphic to G. A well-known algorithm of Alon, Yuster and Zwick (1995) efficiently reduces this to the "colored" version of the problem, denoted G-SUB, and then solves G-SUB in time $O(n^{tw(G)+1})$ where $tw(G)$ is the treewidth of G. Marx (2010) conjectured that G-SUB requires time $\Omega(n^{\text{const}\cdot tw(G)})$ and, assuming the Exponential Time Hypothesis, proved a lower bound of $\Omega(n^{\text{const}\cdot emb(G)})$ for a certain graph parameter $emb(G) = \Omega(tw(G)/\log tw(G))$. With respect to the size of AC^0 circuits solving G-SUB, Li, Razborov and Rossman (2017) proved an unconditional average-case lower bound of $\Omega(n^{\kappa(G)})$ for a different graph parameter $\kappa(G) = \Omega(tw(G)/\log tw(G))$.

Our contributions are as follows. First, we show that $emb(G)$ is at most $O(\kappa(G))$ for all graphs G. Next, we show that $\kappa(G)$ can be asymptotically less than $tw(G)$; for example, if G is a hypercube then $\kappa(G)$ is $\Theta\left(tw(G)/\sqrt{\log tw(G)}\right)$. Finally, we construct AC^0 circuits of size $O(n^{\kappa(G)+\text{const}})$ that solve G-SUB in the average case, on a variety of product distributions. This improves an $O(n^{2\,\kappa(G)+\text{const}})$ upper bound of Li et al., and shows that the average-case complexity of G-SUB is $n^{o(tw(G))}$ for certain families of graphs G such as hypercubes.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Fixed parameter tractability; Mathematics of computing → Graph algorithms

Keywords and phrases subgraph isomorphism, average-case complexity, AC^0, circuit complexity

Digital Object Identifier 10.4230/LIPIcs.IPEC.2019.24

Related Version The full paper is available at https://arxiv.org/abs/1902.06380.

Funding *Gregory Rosenthal*: NSERC (PGS D)

Acknowledgements Thanks to Benjamin Rossman for introducing me to this topic, and for having many helpful discussions about the research and about drafts of this paper. Thanks to Henry Yuen for providing feedback on a draft of this paper as well. Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing.

1 Introduction

The subgraph isomorphism problem asks, given graphs X and G, whether X has a subgraph isomorphic to G. In the "colored" or "partitioned" version of the problem, each vertex of the larger graph X comes with a "color" from the vertex set of G, and we ask whether X has a subgraph that is isomorphic to G with respect to this coloring. We denote the uncolored and colored subgraph isomorphism problems by $G\text{-SUB}_{\text{uncol}}(X)$ and $G\text{-SUB}(X)$ respectively.

Subgraph isomorphism is NP-complete (e.g. if G is a clique or Hamiltonian cycle), so research has focused on algorithms for a variety of special cases in the context of parameterized complexity, surveyed in [12]. If G is a fixed graph on k vertices then $G\text{-SUB}_{\text{uncol}}$ is solvable in time $O(n^k)$ by brute force, where (here and throughout this section) n is the order of the input graph. The color-coding algorithm of Alon, Yuster and Zwick [2] improves on this by efficiently reducing $G\text{-SUB}_{\text{uncol}}$ to G-SUB and solving the latter in time $O(n^{tw(G)+1})$, where $tw(G)$ is the treewidth of the fixed graph G.

The exponent $tw(G)+1$ can sometimes be improved using fast matrix multiplication [14, 5], but no significantly faster algorithm is known for either the colored or uncolored subgraph isomorphism problem. The following was conjectured in [11]:

▶ **Conjecture 1.1.** *There is no class $\mathcal{G}$ of graphs with unbounded treewidth, no algorithm $\mathbb{A}$ that on inputs G and X solves G-SUB(X), and no function f such that if G is in $\mathcal{G}$ then $\mathbb{A}$ runs in time $f(G)n^{o(tw(G))}$.*

Marx [11] came close to proving Conjecture 1.1 assuming the Exponential Time Hypothesis (ETH) [9], which is the hypothesis that solving 3SAT on n variables requires $2^{\Omega(n)}$ time. We state his result in terms of a parameter $emb(G)$ (short for "embedding") which we will define in Section 4:

▶ **Theorem 1.2** ([11])**.** *If there is a class $\mathcal{G}$ of graphs with unbounded treewidth, an algorithm $\mathbb{A}$ that on inputs G and X solves G-SUB(X), and a function f such that if G is in $\mathcal{G}$ then $\mathbb{A}$ runs in time $f(G)n^{o(emb(G))}$, then ETH is false.*

Marx [11] proved that $emb(G)$ is $\Omega(tw(G)/\log tw(G))$, so Theorem 1.2 comes within a logarithmic factor in the exponent of proving Conjecture 1.1. Our main result is a counterexample to an average-case analogue of Conjecture 1.1, in a sense that will be made precise in Section 3. Moreover, our result holds on circuits of depth depending only on G.

Li, Razborov and Rossman [10] proved that for fixed G, the average-case AC^0 complexity of G-SUB is between $n^{\kappa(G)-o(1)}$ and $n^{2\,\kappa(G)+c}$, where $\kappa(G)$ is a graph property defined in Section 3 and c is an absolute constant.[1] We tighten this gap, answering an open problem posed in [10]:

▶ **Theorem 1.3.** *There is a constant $c > 0$ such that for any fixed graph G, the average-case AC^0 complexity of G-SUB is at most $n^{\kappa(G)+c}$.*

We observe that a similar result holds easily on Turing machines, using as a subroutine the *sort-merge join* algorithm from relational algebra. This involves sorting, which cannot be done in AC^0 [7], so our circuit instead uses hashing that relies on concentration of measure for subgraphs of random graphs.

It was also proved in [10] that $\kappa(G)$ is between $\Omega(tw(G)/\log tw(G))$ and $tw(G)+1$, from which it follows that the *worst*-case complexity of G-SUB on bounded-depth circuits is at least $n^{\Omega(tw(G)/\log tw(G))}$. The question was posed in [10] of whether $\kappa(G)$ is $\Theta(tw(G))$; an affirmative answer would have implied that Conjecture 1.1 holds on bounded-depth circuits.

Our main result is a separation of κ from treewidth. The Hamming graph K_q^d has vertex set $\{1,\ldots,q\}^d$ and edges between every two vertices that differ in exactly one coordinate. It is already known that K_q^d has treewidth $\Theta\left(q^d/\sqrt{d}\right)$ [4]. We prove the following:

▶ **Theorem 1.4.** *$\kappa\left(K_q^d\right)$ is $\Theta(q^d/d)$.*

Thus, if G is the hypercube graph K_2^d for example, then $\kappa(G)$ is $\Theta\left(tw(G)/\sqrt{\log tw(G)}\right)$. It follows that an average-case analogue of Conjecture 1.1 is false if $\mathcal{G}$ is taken to be the set of all hypercubes. We also prove the following (for arbitrary graphs G):

▶ **Theorem 1.5.** *$emb(G)$ is $O(\kappa(G))$.*

[1] In [10], the parameter $\kappa(G)$ was called $\kappa_{\mathrm{col}}(G)$.

Because of Theorem 1.5, even if our upper bound generalizes to the worst case, it is still consistent with current knowledge (in particular Theorem 1.2) that ETH is true. Another consequence of Theorem 1.5 is that the lower bound from Theorem 1.2 holds unconditionally in AC^0.

It follows from Theorems 1.4 and 1.5 that if G is a hypercube then $emb(G) \leq O(\kappa(G)) = o(tw(G))$, so proving that Conjecture 1.1 holds under ETH cannot be done by proving that $emb(G)$ is $\Theta(tw(G))$. In fact, this conclusion was already known: Alon and Marx [1] proved that if G is a 3-regular expander then $emb(G)$ is $\Theta(tw(G)/\log tw(G))$. It was proved in [10] that if G is a 3-regular expander then $\kappa(G)$ is $\Theta(tw(G))$, which makes our separation of κ from treewidth more surprising. On the other hand, we will see that Theorem 1.5 is asymptotically tight in the case of Hamming graphs.

We can make a similar statement regarding AC^0. Amano [3] observed that the color-coding algorithm for G-SUB can be implemented by AC^0 circuits of size $O(n^{tw(G)+1})$ for fixed G. Our separation of κ from treewidth implies that if Conjecture 1.1 holds in AC^0, then this cannot be proved using average-case complexity as defined here and in [10].

The paper is organized as follows. In Section 2 we introduce some notation and definitions. In Section 3 we define the average-case problem and $\kappa(G)$, and give an $\tilde{O}(n^{\kappa(G)})$-time algorithm for the average-case problem. In Section 4 we define $emb(G)$ and prove that $emb(G)$ is $O(\kappa(G))$. In Section 5 we prove that $\kappa\left(K_q^d\right)$ is $\Theta(q^d/d)$, and obtain as a corollary that $emb\left(K_q^d\right)$ is $\Theta(q^d/d)$ as well. We also summarize the proof of Chandran and Kavitha [4] that $tw\left(K_q^d\right)$ is $\Theta\left(q^d/\sqrt{d}\right)$. In Section 6 we prove our AC^0 upper bound.

2 Preliminaries

It will be convenient to define $\tilde{O}(f(n)) = f(n)\log^{O(1)} n$. (This differs from the standard notation when $f(n) = n^{o(1)}$.) Let $[k] = \{1, \ldots, k\}$ for $k \in \mathbb{N}$.

We use **boldface** to denote random variables. The indicator variable $\mathbb{1}\{E\}$ equals 1 if the event E occurs and 0 otherwise. Expected value is denoted $\mathbb{E}[\cdot]$. An event occurs *asymptotically almost surely (a.a.s.)* if it occurs with probability $1 - o(1)$ as n goes to infinity.

2.1 Graphs

All graphs we consider are simple and undirected, and may have isolated vertices. If G is a graph then let $V(G)$ and $E(G)$ denote its vertex and edge sets, with respective cardinalities $v(G)$ and $e(G)$. If u and v are adjacent vertices then we denote the edge connecting them by uv or vu. A graph H is a subgraph of G, denoted $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

▶ **Definition 2.1** (Colored subgraph isomorphism problem). *For graphs G and X, where X comes with a coloring $\chi : V(X) \to V(G)$, the problem G-SUB(X) asks whether X has a subgraph G' such that χ (restricted to $V(G')$) is an isomorphism from G' to G. (Note that G' is not required to be an induced subgraph of X.)*

For $U \subseteq V(G)$ let $G[U]$ be the induced subgraph of G on U, and more generally let $G[U_1, \ldots, U_k] = G[U_1 \cup \cdots \cup U_k]$. Let $G - U = G[V(G) - U]$, and for $H \subseteq G$ let $G - H = G - V(H)$.

When the parent graph G is clear in context, let $\deg(u)$ be the degree of a vertex u, and for disjoint $S, T \subseteq V(G)$ let $e(S, T)$ be the number of edges between S and T. Similarly, for vertex-disjoint graphs A and B let $e(A, B) = e(V(A), V(B))$.

Let $G \cap H$ be the graph with vertex set $V(G) \cap V(H)$ and edge set $E(G) \cap E(H)$, and define $G \cup H$ similarly. Note that $G \cap H$ may have isolated vertices even if G and H do not. If $A \subseteq B$ are graphs then let $[A, B] = \{H \mid A \subseteq H \subseteq B\}$, and let $(A, B]$ be the same interval without A, etc.

We denote by K_k the complete graph on k vertices, also called the k-clique. The graph K_q^d has vertex set $[q]^d$, and two vertices are adjacent if and only if they differ in exactly one coordinate. Such graphs are called *Hamming graphs*. A special case is the d-dimensional hypercube $Q_d = K_2^d$; we will use $\{0,1\}^d$ for its vertex set.

Finally, let $\mathbf{ER}(n, p)$ be the Erdős-Rényi graph on n vertices in which each possible edge exists independently with probability p.

3 The Average-Case Problem and the Parameter $\kappa(G)$

3.1 Threshold Random Graphs

First we will define *threshold weightings*, which assign weights to the vertices and edges of a graph subject to certain constraints. Then we will define a family of random graphs for each threshold weighting. The content in this subsection is essentially all from [10].

▶ **Definition 3.1.** *A threshold weighting on a graph G is a pair $(\alpha, \beta) \in [0,1]^{V(G)} \times [0,2]^{E(G)}$ with the following property. For $H \subseteq G$ let $\alpha(H) = \sum_{u \in V(H)} \alpha(u)$ and $\beta(H) = \sum_{e \in E(H)} \beta(e)$, and let $\Delta(H) = \alpha(H) - \beta(H)$. Then, $\Delta(H) \geq 0$ for all $H \subseteq G$, and $\Delta(G) = 0$. Let $\theta(G)$ be the set of threshold weightings on G.*

We will often denote $\Delta = (\alpha, \beta)$ in a slight abuse of notation. (Since $\Delta(u) = \alpha(u)$ if u is a single vertex, the pair (α, β) is uniquely determined by Δ.) The requirement that α be nonnegative is redundant because it's a special case of the requirement that Δ be nonnegative. The requirement that $\beta \leq 2$ is also redundant because for every edge uv,

$$0 \leq \Delta(uv) = \alpha(u) + \alpha(v) - \beta(uv) \leq 2 - \beta(uv).$$

A trivial example is $(\alpha, \beta) = (0, 0)$, i.e. all vertices and edges have a weight of zero. The following example is more general:

▶ **Example 3.2** (Markov Chains)**.** Let $M \in \mathbb{R}_{\geq 0}^{V(G) \times V(G)}$ be a column stochastic matrix (meaning each column sums to 1) such that if $M_{u,v} \neq 0$ then either $u = v$ or $uv \in E(G)$. Let $\alpha(u) = 1 - M_{u,u}$ for all u, and $\beta(uv) = M_{u,v} + M_{v,u}$ for all $uv \in E(G)$. Then for all $H \subseteq G$,

$$\Delta(H) = \sum_{\substack{v \in V(H) \\ uv \in E(G) - E(H)}} M_{u,v} \geq 0, \tag{1}$$

with equality if $H = G$. In fact, in the full paper we prove that every threshold weighting is equivalent to at least one Markov Chain.

The following threshold weighting will be especially important, and can be thought of as representing a uniform random walk on G:

▶ **Definition 3.3.** *If G lacks isolated vertices then let $\Delta_o = (1, \beta_o) \in \theta(G)$ be the threshold weighting generated in Example 3.2 when $M_{u,v} = \mathbb{1}\{uv \in E(G)\}/\deg(v)$. That is, $\Delta_o = (\alpha, \beta)$, where $\alpha(u) = 1$ for all u and $\beta(uv) = 1/\deg(u) + 1/\deg(v)$ for all $u \neq v$. If G is d-regular then this simplifies to $\Delta_o = (1, \beta_o) = (1, 2/d)$.*

Now we define threshold random graphs:

▶ **Definition 3.4.** *For $\Delta = (\alpha, \beta) \in \theta(G)$ let $\mathbf{X}_{\Delta,n}$ be the graph with vertices u_i for $u \in V(G)$ and $i \in [n^{\alpha(u)}]$, and for $uv \in E(G)$, each edge $u_i v_j$ independently with probability $n^{-\beta(uv)}$. The graph $\mathbf{X}_{\Delta,n}$ comes with the coloring to G defined by $u_i \mapsto u$.*

For $H \subseteq G$ and X in the support of $\mathbf{X}_{\Delta,n}$, let $\mathrm{Sub}_X(H)$ be the set of subgraphs $H' \subseteq X$ such that the aforementioned coloring (restricted to $V(H')$) is an isomorphism from H' to H. We say that such a graph H' is "H-colored". Note that $\mathrm{Sub}_X(H)$ can be identified with a subset of $\prod_{u \in V(H)} [n^{\alpha(u)}]$.

▶ **Lemma 3.5.** *If $\Delta \in \theta(G)$ and $H \subseteq G$ then $\mathbb{E}[|\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(H)|] = n^{\Delta(H)}(1 \pm o(1))$.*

Proof. Let $(\alpha, \beta) = \Delta$. The set $\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(H)$ contains each of its $n^{\alpha(H)}$ possible elements with probability $n^{-\beta(H)}$, so the result follows from linearity of expectation. (The $1 \pm o(1)$ accounts for having to round $n^{\alpha(\cdot)}$ to an integer.) ◀

Lemma 3.5 motivates the requirements that Δ be nonnegative everywhere and that $\Delta(G) = 0$. Recall that the problem G-SUB(X) asks whether $\mathrm{Sub}_X(G)$ is the empty set. Since $\Delta(G)$ is required to be zero, it follows that $\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(G)$ has (approximately) one element on average, and the probability that $\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(G)$ is empty is known to be bounded away from 0 and 1 as n goes to infinity [10].

3.2 The Parameter $\kappa(G)$ and an Algorithm for the Average Case

We now define $\kappa(G)$:

▶ **Definition 3.6** ([10])**.** *Let G be a graph with no isolated vertices. Let $\mathrm{Seq}(G)$ be the set of* union sequences*, meaning sequences $(H_1, \ldots, H_k)$ of distinct subgraphs of G such that $H_k = G$ and each H_i is either an edge or the union of two previous graphs in the sequence. For $\Delta \in \theta(G)$ let $\kappa_\Delta(G) = \min_{S \in \mathrm{Seq}(G)} \max_{H \in S} \Delta(H)$. Finally, let $\kappa(G) = \max_{\Delta \in \theta(G)} \kappa_\Delta(G)$.*

To simplify the exposition, whenever we refer to $\kappa(G)$, the graph G is implicitly assumed to lack isolated vertices. It was proved in [10] that for any fixed G, constant-depth circuits solving G-SUB$(\mathbf{X}_{\Delta,n})$ a.a.s. require size at least $n^{\kappa_\Delta(G) - o(1)}$ and at most $n^{2\kappa_\Delta(G)+c}$ (where c is an absolute constant). The results about average-case complexity described in Section 1 are with respect to a Δ such that $\kappa_\Delta(G) = \kappa(G)$.

▶ **Theorem 3.7.** *The problem G-SUB$(\mathbf{X}_{\Delta,n})$ can be solved in time $\tilde{O}(n^{\kappa_\Delta(G)}) \leq \tilde{O}(n^{\kappa(G)})$ a.a.s. for any fixed G.*

Proof. First we prove a weaker upper bound of $\tilde{O}(n^{2\kappa_\Delta(G)})$, in a manner analogous to the circuit from [10], and then we describe a modification (on Turing machines) that removes the factor of 2 from the exponent. In Section 6 we will remove the factor of 2 in AC^0 using a different approach.

Let S be a union sequence such that $\kappa_\Delta(G) = \max_{H \in S} \Delta(H)$. For any $H \in S$, by Lemma 3.5 and Markov's Inequality, $P\left(|\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(H)| > n^{\Delta(H)} \log n\right) \leq 1/\log n$. (We will obtain a tighter bound of $P(|\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(H)| > \tilde{O}(n^{\Delta(H)})) \leq n^{-\omega(1)}$ in Section 6.1.) By a union bound it follows that if $X \sim \mathbf{X}_{\Delta,n}$ then $\max_{H \in S} |\mathrm{Sub}_X(H)| \leq \tilde{O}(n^{\kappa_\Delta(G)})$ a.a.s. Assume this condition holds for X. For each successive H in S, compute $\mathrm{Sub}_X(H)$ as follows. If H is a single edge then this is trivial. Otherwise $H = A \cup B$ for some previous $A, B \in S$, in which case $\mathrm{Sub}_X(H)$ is the set of $\mathcal{A} \cup \mathcal{B}$ such that $\mathcal{A} \in \mathrm{Sub}_X(A), \mathcal{B} \in \mathrm{Sub}_X(B)$ and the projections of $\mathcal{A}$ and $\mathcal{B}$ onto $[n]^{V(A \cap B)}$ are equal. Therefore $\mathrm{Sub}_X(H)$ can be computed by brute force in time $\tilde{O}(|\mathrm{Sub}_X(A)| \cdot |\mathrm{Sub}_X(B)|) \leq \tilde{O}(n^{2\kappa_\Delta(G)})$. Finally, check whether $\mathrm{Sub}_X(G)$ is empty.

We can save a quadratic factor by computing $\mathrm{Sub}_X(H)$ from $\mathrm{Sub}_X(A)$ and $\mathrm{Sub}_X(B)$ as follows. (This is a case of the *sort-merge join* algorithm for computing the *natural join* of two relations, as defined in database theory [20].) Fix an efficiently computable total order on $[n]^{V(A\cap B)}$, e.g. interpret elements of $[n]^{V(A\cap B)}$ as $v(A\cap B)$-digit base-n numbers in increasing order, and then define a partial order on $[n]^{V(A)} \cup [n]^{V(B)}$ by first projecting onto $[n]^{V(A\cap B)}$. Sort $\mathrm{Sub}_X(A)$ and $\mathrm{Sub}_X(B)$ in nondecreasing order, and for convenience add the symbol $\perp$ to the end of both sorted lists. Let $\mathcal{A}$ and $\mathcal{B}$ be the first elements of $\mathrm{Sub}_X(A)$ and $\mathrm{Sub}_X(B)$ respectively, and initialize an empty accumulator (which will ultimately equal $\mathrm{Sub}_X(H)$). While $\mathcal{A} \neq \perp$ and $\mathcal{B} \neq \perp$, do the following. If $\mathcal{A} < \mathcal{B}$ then let $\mathcal{A}$ be the next element of $\mathrm{Sub}_X(A)$. If $\mathcal{B} < \mathcal{A}$ then let $\mathcal{B}$ be the next element of $\mathrm{Sub}_X(B)$. Otherwise, let $\mathcal{B}' = \mathcal{B}$, and while $\mathcal{B}' \neq \perp$ and the projections of $\mathcal{A}$ and $\mathcal{B}'$ onto $[n]^{V(A\cap B)}$ are equal, add $\mathcal{A} \cup \mathcal{B}'$ to the accumulator and let $\mathcal{B}'$ be the next element of $\mathrm{Sub}_X(B)$. Then (once the procedure involving $\mathcal{B}'$ has finished) let $\mathcal{A}$ be the next element of $\mathrm{Sub}_X(A)$.

Sorting $\mathrm{Sub}_X(A)$ and $\mathrm{Sub}_X(B)$ takes $\tilde{O}(|\mathrm{Sub}_X(A)| + |\mathrm{Sub}_X(B)|)$ comparisons, and then computing $\mathrm{Sub}_X(H)$ takes $\tilde{O}(|\mathrm{Sub}_X(A)| + |\mathrm{Sub}_X(B)| + |\mathrm{Sub}_X(H)|) \leq \tilde{O}(n^{\kappa_\Delta(G)})$ time. ◀

We will use the following graph-theoretic properties of $\kappa(G)$:

▶ **Theorem 3.8** ([10][2]). *Let G be a graph with no isolated vertices.*

- **(i)** *There exists $\Delta = (1, \beta) \in \theta(G)$ (meaning $\Delta(u) = 1$ for all vertices u) such that $\kappa(G) = \kappa_\Delta(G)$.*
- **(ii)** *$\kappa(G) \geq v(G)h(G)/(3\max_{u\in V(G)} \deg(u))$, where $h(G)$ is the edge expansion of G.*
- **(iii)** *If G is a minor of some graph H then $\kappa(G) \leq \kappa(H)$.*

The following was observed in [10] as well:

▶ **Corollary 3.9.** *If G is a bounded-degree expander then $\kappa(G)$ is $\Theta(v(G))$.*

Proof. Theorem 3.8(ii) implies that $\kappa(G)$ is $\Omega(v(G))$. Recall from Section 1 that $\kappa(G) \leq tw(G) + 1$ [10], and it is well known that $tw(G) + 1 \leq v(G)$. ◀

4 The Parameter $emb(G)$ and Proof that $emb(G)$ is $O(\kappa(G))$

Recall that $emb(G)$ is significant because of its role in Marx's ETH-hardness result for G-SUB, namely Theorem 1.2.

▶ **Definition 4.1** ($emb(G)$). *Let $G^{(q)}$ be the graph formed by replacing each vertex of G with a q-clique, i.e. it has vertices u_i for all $u \in V(G)$ and $i \in [q]$, and edges u_iv_j for all $u_i \neq v_j$ such that either $u = v$ or $uv \in E(G)$. Let $emb(G)$ be the supremum of all $r > 0$ for which there exists $m_0 = m_0(G, r)$ such that if H is any graph with $m \geq m_0$ edges and no isolated vertices, then H is a minor of $G^{(\lceil m/r\rceil)}$, and furthermore a minor mapping from H to $G^{(\lceil m/r\rceil)}$ can be computed in time $f(G)m^{O(1)}$ for some function f.*

Although the requirement that such a minor mapping be efficiently computable is crucial in Theorem 1.2, none of the other results about $emb(G)$ that we reference or derive depend on this requirement, so we may safely ignore it going forward. The following example illustrates Definition 4.1:

[2] Specifically, Corollary 4.2, Theorem 4.9, and Theorem 5.1 of [10] correspond to Theorems 3.8(i) to 3.8(iii) respectively.

▶ **Example 4.2** ($emb(K_k)$ [11]). Since $K_k^{(\lceil m/r \rceil)} = K_{k\lceil m/r \rceil}$, any graph H with m edges is a minor of $K_k^{(\lceil m/r \rceil)}$ if and only if $v(H) \leq k\lceil m/r \rceil$. If H has no isolated vertices then H could have up to $2m$ vertices, so $2m \leq k\lceil m/r \rceil$. Therefore $emb(K_k) = k/2$: it is sufficient for $2m$ to be at most km/r (i.e. $r \leq k/2$), and no $r > k/2$ satisfies $2m \leq k\lceil m/r \rceil$ for arbitrarily large m.

▶ Remark. The name $emb(G)$ comes from the fact that Marx [11] called a minor mapping from H to $G^{(q)}$ an "embedding of depth q" from H into G. Marx [11] used the notation $G^{(q)}$, but the parameter $emb(G)$ is new in the current paper, all results about $emb(G)$ in [11, 1] having been stated in terms of embeddings of some depth.

The following is used in our proof that $emb(G)$ is $O(\kappa(G))$:

▶ **Lemma 4.3.** $\kappa\left(G^{(q)}\right) \leq q \max(\kappa(G), 2)$.

Proof Sketch. Given a threshold weighting Δ on $G^{(q)}$, collapsing each cluster of q vertices to a single "mega-vertex" induces a threshold weighting Δ' on G. Let S be an optimal union sequence for G with respect to Δ', and project S back onto $G^{(q)}$. ◀

Now we prove that $emb(G)$ is $O(\kappa(G))$ (Theorem 1.5), using an argument similar to the proof by Marx [11] that $emb(G)$ is $O(tw(G))$:

Proof. Let $r > 0$, and assume there exists an arbitrarily large 3-regular expander H that's a minor of $G^{(\lceil e(H)/r \rceil)}$. Then by Corollary 3.9, Theorem 3.8(iii), and Lemma 4.3,

$$e(H) = \Theta(v(H)) = \Theta(\kappa(H)) \leq O\left(\kappa\left(G^{(\lceil e(H)/r \rceil)}\right)\right) \leq O\left(\kappa(G)e(H)/r\right),$$

so r must be $O(\kappa(G))$. ◀

In [10] the question was posed of whether Theorem 1.2 holds with $\kappa(G)$ in place of $emb(G)$. By Theorem 1.5 this would be a stronger bound, which makes the question even more interesting. This problem is open even in the case of 3-regular expanders: recall from Section 1 that if G is a 3-regular expander then $emb(G)$ is $\Theta(tw(G)/\log tw(G))$ and $\kappa(G)$ is $\Theta(tw(G))$ [1, 10].

The fact that $\kappa(G)$ is $\Omega(emb(G))$ gives an alternate proof, besides the one in [10], that $\kappa(G)$ is $\Omega(tw(G)/\log tw(G))$.

5 Separating κ from Treewidth

In Section 5.1 we prove that $\kappa(K_k) = k/4 + O(1)$, which is a special case of the more general result that $\kappa\left(K_q^d\right) = \Theta(q^d/d)$. We obtain tighter multiplicative constants in the case $d = 1$, and it provides an opportunity to illustrate the main ideas of our proof in a simpler setting, but when reading the full paper it may be skipped without penalty. In Section 5.2 we prove that $\kappa\left(K_q^d\right)$ is $O(q^d/d)$ when q is even, which is sufficient to separate κ from treewidth. Again, this case is cleaner than the general case and conveys most of the intuition behind it. In an appendix in the full paper we prove that $\kappa\left(K_q^d\right)$ is $O(q^d/d)$ for all q. In Section 5.3 we prove that $\kappa\left(K_q^d\right)$ is $\Omega(q^d/d)$ in two different ways, completing the proof that $\kappa\left(K_q^d\right)$ is $\Theta(q^d/d)$ (Theorem 1.4), and we obtain as a corollary that $emb\left(K_q^d\right)$ is $\Theta(q^d/d)$ as well. In Section 5.4 we summarize the proof of Chandran and Kavitha [4] that $tw\left(K_q^d\right)$ is $\Theta\left(q^d/\sqrt{d}\right)$.

5.1 Proof that $\kappa(K_k) = k/4 + O(1)$

▶ Remark. It was already observed in [10] that $\kappa(K_k)$ is $\Theta(k)$.

Rossman [16] proved that $\kappa_{\Delta_o}(K_k) \geq k/4$, so it suffices to prove the upper bound. By Theorem 3.8(i) it suffices to prove that $\kappa_{\Delta}(K_k) \leq k/4 + O(1)$ for an arbitrary $\Delta = (1, \beta) \in \theta(G)$. First we construct a sequence $U_1 \subseteq \cdots \subseteq U_k = V(K_k)$ such that U_i is an i-element subset of $V(K_k)$, and $\beta(K_k[U_i]) \geq \beta_o(K_k[U_i])$ for all i. The set $U_k = V(K_k)$ satisfies this requirement because $\beta(K_k)$ and $\beta_o(K_k)$ are both equal to k. Given U_i, let $\mathbf{U}_{i-1}$ be an $(i-1)$-element subset of U_i chosen uniformly at random. Each pair of elements in U_i is included in $\mathbf{U}_{i-1}$ with the same probability p_i $(= 1 - 2/i)$, so it follows from linearity of expectation that

$$\mathbb{E}[\beta(K_k[\mathbf{U}_{i-1}])] = \sum_{e \in E(K_k[U_i])} \beta(e) p_i = p_i \beta(K_k[U_i]) \geq p_i\, \beta_o(K_k[U_i]) = \mathbb{E}[\beta_o(K_k[\mathbf{U}_{i-1}])].$$

Therefore there exists a fixed U_{i-1} such that $\beta(K_k[U_{i-1}]) \geq \beta_o(K_k[U_{i-1}])$.

We construct a union sequence S for K_k as follows. Start by enumerating the edges, and then for i from 1 to $k-1$, append $(K_k[U_i] \cup e_1, K_k[U_i] \cup e_1 \cup e_2, \ldots, K_k[U_{i+1}])$, where $e_1, e_2, \ldots$ are the edges between U_i and $U_{i+1} - U_i$. Then,

$$\max_{H \in S} \Delta(H) \leq \max_i \Delta(K_k[U_i]) + 1 \leq \max_i \Delta_o(K_k[U_i]) + 1.$$

As observed in [16], it follows from Equation (1) that $\Delta_o(K_k[U_i]) = i(k-i)/k$, which is at most $k/4$ (when $i = k/2$). Therefore $\kappa_{\Delta}(K_k) \leq k/4 + 1$.

5.2 Proof that $\kappa\left(K_q^d\right)$ is $O(q^d/d)$ if q is Even

First we reduce this to the case $q = 2$. The graph K_q^d is a subgraph of $Q_d^{\left((q/2)^d\right)}$ (recall Definition 4.1), as explained in the full paper. By Theorem 3.8(iii) and Lemma 4.3, if $\kappa(Q_d)$ is $O(2^d/d)$ then

$$\kappa\left(K_q^d\right) \leq \kappa\left(Q_d^{\left((q/2)^d\right)}\right) \leq O\left(\left(\frac{q}{2}\right)^d \kappa(Q_d)\right) \leq O\left(\left(\frac{q}{2}\right)^d \frac{2^d}{d}\right) = O(q^d/d).$$

Now we prove that $\kappa(Q_d)$ is $O(2^d/d)$, following some brief definitions and a high-level overview of the argument. Fix d. We identify each $u \in \{0,1\}^d$ with $\sum_{i=0}^{d-1} u_i 2^i$. For $0 \leq a \leq 2^d$ let $G(a) = Q_d[0, \ldots, a-1]$. Recall that $\Delta_o = (1, \beta_o) = (1, 2/d)$ is a threshold weighting on Q_d (Definition 3.3). Let $\mu = \max_{0 \leq a \leq 2^d} \Delta_o(G(a))$.

▶ Remark. The intuition behind μ is as follows. The reader may note that $\kappa_{\Delta_o}(Q_d) \leq \mu + 1$, by reasoning analogous to that in Section 5.1. That is, for each vertex u of Q_d in increasing lexicographic order, add to an accumulator all edges uv for which $v < u$.

There is another union sequence captured by μ as well. If a subgraph $B \subseteq Q_d$ isomorphic to Q_k for some k, then since Q_k is isomorphic to $G(2^k)$ (and β_o is uniform) it follows that $\Delta_o(B) \leq \mu$. Consider a depth-d binary tree in which each node at depth k is a subgraph of Q_d isomorphic to Q_{d-k} (in particular, the root is Q_d and the leaves are vertices), and each interior node is the union of its two children along with some additional edges corresponding to a coordinate cut. This tree describes a union sequence S for Q_d: recursively obtain the graphs L and R corresponding to the children of Q_d, and then take $L \cup R$ and add the missing edges. Note that $\max_{H \in S} \Delta_o(H) = 2 \max_{0 \leq k \leq d} \Delta_o(G(2^k)) \leq 2\mu$.

Analogous to Section 5.1, the upper bound is obtained by comparing $\kappa_\Delta(Q_d)$ to μ for each Δ, and bounding μ. For this purpose we will consider the two union sequences mentioned above, as well as hybrids of them.

In the full paper we prove that $\kappa(Q_d)$ is $O(\mu)$. It follows from Equation (1) that $\mu = \max_a \Delta_{\text{o}}(G(a)) = \max_a e(G(a), Q_d - G(a))/d$, and in the full paper we prove that $\max_a e(G(a), Q_d - G(a))$ is $O(2^d)$.

5.3 Proof that $\kappa\left(K_q^d\right)$ is $\Omega(q^d/d)$ and $emb\left(K_q^d\right)$ is $\Theta(q^d/d)$

Alon and Marx [1, Theorem 4.3] proved that $emb\left(K_q^d\right)$ is $\Omega(q^d/d)$, and it follows from Theorem 1.5 that $emb\left(K_q^d\right) \leq O\left(\kappa\left(K_q^d\right)\right) \leq O(q^d/d)$. Therefore $emb\left(K_q^d\right)$ is $\Theta(q^d/d)$.

It is implicit in the above argument that $\kappa\left(K_q^d\right) \geq \Omega\left(emb\left(K_q^d\right)\right) \geq \Omega(q^d/d)$. In the full paper we present a second proof that $\kappa\left(K_q^d\right)$ is $\Omega(q^d/d)$, using Theorem 3.8(ii).

5.4 Proof that $tw\left(K_q^d\right)$ is $\Theta\left(q^d/\sqrt{d}\right)$, Summarized

(See [4] for the full proof.) The proof that $tw\left(K_q^d\right)$ is $O\left(q^d/\sqrt{d}\right)$ reduces to the case $q = 2$ by reasoning analogous to that in the beginning of Section 5.2. For $k \in [d]$ let U_k be the set of vertices of Q_d with exactly k or $k-1$ ones. The path $(U_1, \ldots, U_d)$ is a tree decomposition of Q_d with width approximately $2\binom{d}{d/2}$, and by Stirling's approximation this is $\Theta\left(2^d/\sqrt{d}\right)$.[3]

For a graph G let $\phi(G)$ be the minimum over all $U \subseteq V(G), v(G)/4 \leq |U| \leq v(G)/2$ of the number of vertices in $V(G) - U$ with at least one neighbor in U. From a result of Robertson and Seymour [15] it follows that $tw(G) \geq \phi(G) - 1$, and from a result of Harper [6] it follows that $\phi\left(K_q^d\right)$ is $\Omega\left(q^d/\sqrt{d}\right)$. (Also note the parallels between $tw(G) \geq \phi(G) - 1$ and Theorem 3.8(ii); interestingly, we've sign that both are tight to within a constant factor in the case of K_q^d.)

6 AC^0 Upper Bound

An AC^0 circuit is a constant-depth circuit with polynomially many unbounded-fanin AND and OR gates and NOT gates. Fix a graph G and threshold weighting $\Delta \in \theta(G)$ for the remainder of this section. We prove the following, which is a more precise statement of Theorem 1.3:

▶ **Theorem 6.1.** *There exists a constant-depth circuit with $n^{\kappa_\Delta(G)+c}$ wires that solves G-SUB$(\mathbf{X}_{\Delta,n})$ with probability $1 - n^{\omega(1)}$, where $c > 0$ is an absolute constant.*

Since in any circuit the number of gates is at most one plus the number of wires, the circuit from Theorem 6.1 has size $n^{\kappa_\Delta(G)+O(1)} \leq n^{\kappa(G)+O(1)}$. (In this discussion, all $\pm O(1)$ terms in an exponent are independent of G.) For comparison, it was proved in [10] (building on a line of previous work [16, 3, 17, 13]) that the average-case AC^0 complexity of G-SUB$(\mathbf{X}_{\Delta,n})$ is between $n^{\kappa_\Delta(G)-o(1)}$ and $n^{2\kappa_\Delta(G)+O(1)}$. Another related result, regarding the uncolored k-clique problem, is that the average-case AC^0 complexity of K_k-SUB$_{\text{uncol}}\left(\mathbf{ER}\left(n, n^{-2/(k-1)}\right)\right)$ is at most $n^{k/4+O(1)}$ [3, 18] $(= n^{\kappa(K_k)\pm O(1)}$ by Section 5.1). See [19] for a survey of the average-case circuit complexity of subgraph isomorphism more generally.

[3] Compared to the tree decomposition from [4], this one is a simpler variant whose width is larger by up to a constant factor.

▶ **Definition 6.2.** *Let X be in the support of $\mathbf{X}_{\Delta,n}$, and let $U \subseteq G$ be an arbitrary graph (which we think of as a "universe"). Let $\mathrm{Sub}_n(U)$ be the set of all possible elements of $\mathrm{Sub}_{\mathbf{X}_{\Delta,n}}(U)$; note that this can be identified with $\prod_{v \in V(U)} [n^{\alpha(v)}]$. If $A \subseteq U$ and $\mathcal{A} \in \mathrm{Sub}_n(A)$ then let $\mathcal{A}$* extend to U in X *if there exists a graph $\mathcal{U} \in \mathrm{Sub}_X(U)$ (called a U-*extension of $\mathcal{A}$*) such that $\mathcal{A} \subseteq \mathcal{U}$. (In context, X or $\mathbf{X}$ will be implicit.) Equivalently, $\mathcal{A}$ could be required to be in $\mathrm{Sub}_X(A)$ rather than $\mathrm{Sub}_n(A)$ in the latter definition.*

Let $\Delta^*_U(A) = \min_{A \subseteq H \subseteq U} \Delta(H)$. Let X be good *if for all graphs $U \subseteq G$ and $A \subseteq U$, and for all $\mathcal{A} \in \mathrm{Sub}_n(A)$ and vertices $v \in V(U) - V(A)$, there are $\tilde{O}\left(n^{\Delta^*_U(A \cup v) - \Delta^*_U(A)}\right)$ values of $i \in [n^{\alpha(v)}]$ such that $\mathcal{A} \cup v_i$ extends to U. (Recall our unconventional definition of $\tilde{O}(\cdot)$ from Section 2, e.g. $\tilde{O}(1)$ denotes $\log^{O(1)} n$.) Finally, let an event occur* with high probability (w.h.p.) *if it occurs with probability $1 - n^{-\omega(1)}$.*

We prove the following:

▶ **Theorem 6.3.** *The graph $\mathbf{X}_{\Delta,n}$ is good w.h.p.*

Observe that this is a substantially stronger concentration bound than the application of Markov's Inequality in the proof of Theorem 3.7. In Section 6.1 we prove Theorem 6.3, and then in Section 6.2 we use this result to prove Theorem 6.1.

6.1 Proof of Theorem 6.3

First we derive some algebraic properties of the threshold weighting Δ.

▶ **Lemma 6.4.** *If $A, B \subseteq G$ then $\Delta(A) + \Delta(B) = \Delta(A \cap B) + \Delta(A \cup B)$.*

Proof. Each vertex or edge in one (resp. two) of A and B is also in one (resp. two) of $A \cap B$ and $A \cup B$. ◀

▶ **Definition 6.5.** *For $A \subseteq U \subseteq G$ let $\Gamma_U(A) = \bigcap\{H \in [A, U] \mid \Delta(H) = \Delta^*_U(A)\}$, and let A be a U-*base *if $\Delta(A) = \Delta^*_U(A)$.*

Throughout this subsection, U will be an arbitrary subgraph of G unless additional structure is imposed on it, and missing subscripts on Δ^* and Γ default to U.

▶ **Lemma 6.6.** *If $A \subseteq U$ then $\Delta(\Gamma(A)) = \Delta^*(A)$ and $A \subseteq \Gamma(A)$.*

Proof. It suffices to show that the set $S = \{H \in [A, U] \mid \Delta(H) = \Delta^*(A)\}$ is closed under intersection. Let $B, C \in S$. By the definition of S, Lemma 6.4, and the fact that $A \subseteq B \cup C$,

$$2\Delta^*(A) = \Delta(B) + \Delta(C) = \Delta(B \cap C) + \Delta(B \cup C) \geq \Delta(B \cap C) + \Delta^*(A),$$

so $\Delta(B \cap C) \leq \Delta^*(A)$. On the other hand, $\Delta(B \cap C) \geq \Delta^*(A)$ because $A \subseteq B \cap C$. Therefore $\Delta(B \cap C) = \Delta^*(A)$, so $B \cap C \in S$. ◀

The proofs of the following two lemmas are of a similar flavor, and are included in the full paper.

▶ **Lemma 6.7.** *If $A \subseteq \Gamma(A) \subseteq U' \subseteq U$ then $\Gamma(A)$ is a U'-base.*

▶ **Lemma 6.8.** *If $A \subseteq B \subseteq U$ then $\Gamma(A) \subseteq \Gamma(B)$.*

We now analyze the concentration of $\mathbf{X}_{\Delta,n}$, making liberal use of the fact that if $n^{O(1)}$ events occur with uniformly high probability then their conjunction also occurs w.h.p. by a union bound. For the rest of this subsection, "extensions" are with respect to an implicit $\mathbf{X} \equiv \mathbf{X}_{\Delta,n}$.

▶ **Lemma 6.9.** *If $A \subseteq U$ and $\Gamma_U(A) = U$ (i.e. $\Delta(H) > \Delta(U)$ for all $H \in [A, U)$) then the number of U-extensions of any $\mathcal{A} \in \mathrm{Sub}_n(A)$ is $\tilde{O}(1)$ w.h.p.*

(The above conditions are equivalent because, by the definition of $\Gamma(A)$, we have $\Gamma(A) = U$ if and only if U is the unique $H \in [A, U]$ that minimizes $\Delta(H)$.)

Proof Sketch. Here we prove the weaker claim that the lemma holds with "a.a.s." in place of "w.h.p." There are $n^{\alpha(U)-\alpha(A)}$ possible U-extensions of $\mathcal{A}$, each of which occurs with probability $n^{-\beta(U)+\beta(A)}$, so $\mathcal{A}$ has $n^{\Delta(U)-\Delta(A)}$ U-extensions in expection. Since $\Delta(U) < \Delta(A)$ by assumption, the result follows from Markov's Inequality. ◀

▶ **Lemma 6.10.** *If A is a U-base then any $\mathcal{A} \in \mathrm{Sub}_n(A)$ has $\tilde{O}(n^{\Delta(U)-\Delta(A)})$ U-extensions w.h.p.*

Proof Sketch. Again, if we replace "w.h.p." with "a.a.s." then the claim follows immediately from Markov's Inequality. A similar lower bound is also proved in an appendix in the full paper. ◀

Now we prove that $\mathbf{X}_{\Delta,n}$ is good w.h.p.:

Proof of Theorem 6.3. Let $A \subseteq U$, $\mathcal{A} \in \mathrm{Sub}_n(A)$ and $v \in V(U) - V(A)$. By a union bound it suffices to prove that w.h.p. there are $\tilde{O}(n^{\Delta^*(A\cup v)-\Delta^*(A)})$ values of i such that $\mathcal{A} \cup v_i$ extends to U. The number of such i is at most the number of i such that $\mathcal{A} \cup v_i$ extends to $\Gamma(A \cup v)$, which is at most the number of $\Gamma(A \cup v)$-extensions of $\mathcal{A}$. Since $\Gamma(A) \subseteq \Gamma(A \cup v)$ (Lemma 6.8), this equals the sum over all $\gamma \in \{\Gamma(A)\text{-extensions of } \mathcal{A}\}$ of the number $\mathbf{E}_\gamma$ of $\Gamma(A \cup v)$-extensions of γ.

It follows from Lemma 6.9 that $\mathcal{A}$ has $\tilde{O}(1)$ extensions to $\Gamma(A)$ w.h.p. (To see this, note that if $A \subseteq H \subset \Gamma(A)$ then $\Delta(H) \geq \Delta^*(A) = \Delta(\Gamma(A))$ (Lemma 6.6), and if $\Delta(H) = \Delta^*(A)$ then it follows from the definition of $\Gamma(A)$ that $\Gamma(A) \subseteq H$, a contradiction.) Since $\Gamma(A)$ is a $\Gamma(A \cup v)$-base (Lemma 6.7), it follows from Lemma 6.10 that any $\mathbf{E}_\gamma$ is $\tilde{O}(n^{\Delta(\Gamma(A\cup v))-\Delta(\Gamma(A))})$ w.h.p. $(= \tilde{O}(n^{\Delta^*(A\cup v)-\Delta^*(A)})$ by Lemma 6.6). ◀

6.2 The Circuit

If D is a data structure then let $|D|$ denote the number of bits used to represent it according to whatever schema we describe. When there is a null element we represent it by the all-zeros string.

Proof of Theorem 6.1. Since $\mathbf{X}_{\Delta,n}$ is good w.h.p. (Theorem 6.3) it suffices to prove the existence of a (small, constant-depth) circuit C such that $P_{X\sim\mathbf{X}_{\Delta,n}}(\mathsf{C}(X) = G\text{-}\mathsf{SUB}(X) \mid X \text{ is good}) = 1 - n^{\omega(1)}$. By Yao's Principle [21] it suffices to prove the existence of a (small, constant-depth) random circuit $\mathbf{C}$ such that $P(\mathbf{C}(X) = G\text{-}\mathsf{SUB}(X)) = 1 - n^{-\omega(1)}$ for any fixed good X.

The following result is essentially implicit in [10] (as is the argument above) and helps keep the random circuit small:

▶ **Lemma 6.11** (Random Hashing)**.** *Let S be a set containing a null element, and assume all elements of S are represented using the same number of bits. Let $l = l(n)$ and $m = m(n)$ be polynomially-bounded functions of n. Then there exists a random, constant-depth circuit $\mathbf{C} : S^l \to S^{\tilde{O}(m)}$ such that if A is an array of l values in S, of which all but at most m are null, then $\mathbf{C}$ has at most $|A|n^{o(1)}$ gates and $|A|\tilde{O}(l/m)$ wires, and w.h.p. the multiset of non-null elements of $\mathbf{C}(A)$ is the same as that of A.*

We remark that Lemma 6.11 will only be called with $l \leq \tilde{O}(n)$.

Proof Sketch. The proof uses a Chernoff bound and a result from [8]. ◀

Given $H \subseteq G$ and an ordering $\pi = (\pi^1, \ldots, \pi^{v(H)})$ of $V(H)$, let $\delta_i = \Delta^*_H(\pi^1 \cup \cdots \cup \pi^i)$ for $0 \leq i \leq v(H)$, and let $\phi_i = \delta_{i+1} - \delta_i$ for $0 \leq i < v(H)$. (In context H and π will be implicit.)

▶ **Lemma 6.12.** *$0 \leq \phi_i \leq 1$ for all i.*

Proof. Clearly $\delta_i \leq \delta_{i+1}$. Let $A \subseteq G$ such that $\pi^1, \ldots, \pi^i \in V(A)$ and $\Delta(A) = \delta_i$. Then $\delta_{i+1} \leq \Delta(A \cup \pi^{i+1}) \leq \Delta(A) + \alpha(\pi^{i+1}) \leq \delta_i + 1$. ◀

Let $T = T(H, \pi)$ be a depth-$v(H)$ tree (i.e. the root has depth 0 and the leaves have depth $v(H)$) in which each node at depth $i < v(H)$ has $n^{\phi_i} \log^{c_i} n$ children, where c_i is a sufficiently large constant. Each non-root node N has a *partial label* $\mathcal{L}(N) \in \{\text{null}\} \cup [n]$, and N's (complete) *label* is the sequence of partial labels along the path from the root to N. A label is considered null if it includes any null partial labels. It is required that no two nodes share a non-null label, and there exists a node labeled with $(l_1, \ldots, l_i)$ if and only if[4] $\{\pi^1_{l_1}, \ldots, \pi^i_{l_i}\}$ extends to H.

Let S be an *immediate subtree* of T (resp. of a node N), denoted $S \in T$ (resp. $S \in N$), if S's root is a child of T's root (resp. of N). Any subtree is considered to have the same (partial) label as its root.

If the underlying tree structure of T (that is, everything except the partial labels) is implicit, then we can represent T by an array of values in $\{\text{null}\} \cup [n]$, indexed by the vertices of T. Each of these values can be associated with a bit string in a natural way. We will consider circuits that compute T according to this representation.

▶ **Lemma 6.13.** *$|T|$ is $\tilde{O}(n^{\Delta(H)})$.*

Proof. $\delta_0 = \Delta(\emptyset) = 0$ and $\delta_{v(H)} = \Delta^*_H(V(H)) = \Delta(H)$. It takes $\tilde{O}(1)$ bits to store an element of $[n]^{V(H)}$, and each ϕ_i is nonnegative (Lemma 6.12), so

$$|T| = \tilde{O}\left(\prod_{i=0}^{v(H)-1} n^{\phi_i}\right) = \tilde{O}\left(n^{\sum_{i=0}^{v(H)-1} \phi_i}\right) = \tilde{O}\left(n^{\delta_{v(H)} - \delta_0}\right) = \tilde{O}\left(n^{\Delta(H)}\right). \quad \blacktriangleleft$$

▶ **Lemma 6.14.** *For all $H \subseteq G$ there exists a random, constant-depth circuit with $\tilde{O}(n^{\Delta(H)+2})$ wires, independent of X, that computes $T(H, \pi')$ from $T(H, \pi)$ w.h.p.*

Proof Sketch. Assume that π and π' differ only in positions d and $d+1$. (The general case can be reduced to at most $\binom{v(H)}{2}$ copies of this circuit in succession.) Define δ'_i and ϕ'_i analogously to δ_i and ϕ_i, but with respect to π' rather than π. Clearly $\delta_i = \delta'_i$ for $i \neq d$, so $\phi_i = \phi'_i$ for $i \notin \{d-1, d\}$.

For each depth-$(d-1)$ node N of $T(H, \pi)$, in parallel, do the following. For $\tau \in N, j \in [n]$ let $A_{\tau j}$ be (if this exists) the subtree rooted at a child of τ whose partial label is j. After updating the partial labels at what will become the new depth-d and depth-$(d+1)$ nodes, hash the number of columns of A down to $\tilde{O}(n^{\phi'_{d-1}})$ (using Lemma 6.11), and hash each remaining column down to a set of $\tilde{O}(n^{\phi'_d})$ elements. The remaining columns are the new immediate subtrees of N, and the remaining elements in each column are now the immediate subtrees of that column. ◀

[4] Recall that $(\pi^j)_{l_j}$ is a π^j-colored vertex in X.

For $e \in E(G)$ we can construct $T(e)$ in a similar manner, as explained in the full paper.

▶ **Lemma 6.15.** *For all $H, H' \subseteq G$ there exists a random, constant-depth circuit, independent of X, with $\tilde{O}(n^{\max(\Delta(H),\Delta(H'))+2})$ wires, that computes $T(H \cup H', \hat{\pi})$ from $T(H, \pi)$ and $T(H', \pi')$ w.h.p. for some $\hat{\pi}$.*

Proof Sketch. Let $T = T(H, \pi)$ and $T' = T(H', \pi')$. By Lemma 6.14 we can assume without loss of generality that $\{\pi^1, \ldots, \pi^{v(H \cap H')}\} = V(H \cap H') = V(H) \cap V(H')$, and that $\pi^k = \pi'^k = \hat{\pi}^k$ for $k \in [v(H \cap H')]$. Define ϕ' and $\hat{\phi}$ with respect to (H', π') and $(H \cup H', \hat{\pi})$ respectively.

Let $\psi_i = \min(\phi_i, \phi'_i)$. For $0 \leq d \leq v(H \cap H')$ let S_d be a depth-d tree in which each node at depth $i < d$ (including $i = 0$) has $\tilde{O}(n^{\psi_i})$ children. Each node of S_d has a (partial) label defined the same way as in T, such that no two nodes share a non-null label, and $\{\pi^1_{l_1}, \ldots, \pi^i_{l_i}\}$ extends to both H and H' (but not necessarily to $H \cup H'$) if and only if some node is labeled with l. Each leaf of S_d with a non-null label l is associated with the pair (τ, τ') of subtrees of T and T' respectively whose labels are also l.

The tree S_0 is the single node (T, T'), and we can compute S_{d+1} from S_d as explained in the full paper. Let $S = S_{v(H \cap H')}$. For d from $v(H \cap H') - 1$ down to 0, for each depth-d node N in S, hash (Lemma 6.11) the number of children of N down from $\tilde{O}(n^{\psi_d})$ to $\tilde{O}(n^{\hat{\phi}_d})$, and if all of N's children are null and $d > 0$ then remove N's partial label. Finally, for each leaf (τ, τ') of S, append a copy of τ' to each leaf of τ, and put this in place of (τ, τ') in S. ◀

For each successive H in an optimal union sequence, compute $T(H)$ as described above, and then apply a single OR gate to all leaves of $T(G)$. ◀

References

1. Noga Alon and Dániel Marx. Sparse balanced partitions and the complexity of subgraph problems. *SIAM J. Discrete Math.*, 25(2):631–644, 2011. doi:10.1137/100812653.
2. Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
3. Kazuyuki Amano. k-subgraph isomorphism on AC^0 circuits. *Comput. Complexity*, 19(2):183–210, 2010. doi:10.1007/s00037-010-0288-y.
4. L. Sunil Chandran and Telikepalli Kavitha. The treewidth and pathwidth of hypercubes. *Discrete Math.*, 306(3):359–365, 2006. doi:10.1016/j.disc.2005.12.011.
5. Friedrich Eisenbrand and Fabrizio Grandoni. On the complexity of fixed parameter clique and dominating set. *Theoret. Comput. Sci.*, 326(1-3):57–67, 2004. doi:10.1016/j.tcs.2004.05.009.
6. L. H. Harper. On an isoperimetric problem for Hamming graphs. *Discrete Appl. Math.*, 95(1-3):285–309, 1999. doi:10.1016/S0166-218X(99)00082-7.
7. Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proc. 18th Ann. ACM Symp. on Theory of Computing*, pages 6–20, 1986. doi:10.1145/12130.12132.
8. Johan Håstad, Ingo Wegener, Norbert Wurm, and Sang-Zin Yi. Optimal depth, very small size circuits for symmetric functions in AC^0. *Inform. and Comput.*, 108(2):200–211, 1994. doi:10.1006/inco.1994.1008.
9. Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.
10. Yuan Li, Alexander Razborov, and Benjamin Rossman. On the AC^0 complexity of subgraph isomorphism. *SIAM J. Comput.*, 46(3):936–971, 2017. doi:10.1137/14099721X.
11. Dániel Marx. Can you beat treewidth? *Theory Comput.*, 6(1):85–112, 2010. doi:10.4086/toc.2010.v006a005.

12 Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask). In *STACS*, volume 25 of *LIPIcs*, pages 542–553. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014. doi:10.4230/LIPIcs.STACS.2014.542.

13 K. Nakagawa and O. Watanabe. Gap Between Two Combinatorial Measures for Constant Depth Circuit Complexity of Subgraph Isomorphism. Technical report, Tokyo Institute of Technology, 2011.

14 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Comment. Math. Univ. Carolin.*, 26(2):415–419, 1985.

15 Neil Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

16 Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proc. 40th Ann. ACM Symp. on Theory of Computing (STOC)*, pages 721–730, 2008. doi:10.1145/1374376.1374480.

17 Benjamin Rossman. *Average-Case Complexity of Detecting Cliques*. Ph.d., MIT, 2010.

18 Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM J. Comput.*, 43(1):256–279, 2014. doi:10.1137/110839059.

19 Benjamin Rossman. Lower bounds for subgraph isomorphism. In *Proc. Intern. Congress of Mathematicians (ICM)*, volume 3, pages 3409–3430, 2018. URL: https://eta.impa.br/dl/051.pdf.

20 Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill Book Company, 6 edition, 2011.

21 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proc. 18th Ann. IEEE Symp. on Foundations of Computer Science*, pages 222–227, 1977. doi:10.1109/SFCS.1977.24.