Electronic Colloquium on Computational Complexity, Report No. 122 (2026)

Complexity of the Graph Homomorphism Problem w.r.t. Degeneracy

Grigorii Braulov * Nikolai Chukhin † Alexander S. Kulikov ‡ Ivan Mihajlin §

Abstract

The graph homomorphism problem HOM is: given an n-vertex source graph G and an h-vertex target graph H, is there a mapping from $V(G)$ to $V(H)$ that preserves edges? A straightforward brute-force algorithm for HOM has running time $O(h^n) = O(2^{n \log h})$ and it is known that, under ETH, there are no $2^{o(n \log h)}$ algorithms. In recent years, less restrictive graph parameters p have been identified that allow one to solve HOM in time $p(H)^{O(n)}$. Examples include treewidth, maximum degree, and track number. These algorithms are faster than $O(h^n)$ and allow one to solve HOM in plain-exponential time $2^{O(n)}$ in the special case when $p(H)$ is bounded. On the other hand, it is known that the chromatic number parameter is too small: under ETH, HOM cannot be solved in time $\chi(H)^{O(n)}$.

We study the complexity of HOM in terms of the *degeneracy* of H. This is perhaps the most natural unresolved graph parameter between the known algorithmic and hardness regimes: on the one hand, each of bounded treewidth, bounded maximum degree, and bounded track number implies bounded degeneracy; on the other hand, bounded degeneracy implies bounded chromatic number. Our results show that, at the same time, the influence of degeneracy of H on the complexity of HOM differs significantly from that of the previously studied parameters. We show that, under ETH, there is no $2^{o(\mathrm{degen}(H) n)}$ algorithm for any value of $\mathrm{degen}(H)$ as a function of n. We also show that bounded degeneracy alone does not make target size benign: even targets with $\mathrm{degen}(H) \leq 2$ and quasi-polynomial size force $n^{\Omega(n)}$-scale hardness. Finally, we introduce a no-compression barrier that explains why the known fine-grained lower bounds for sparse 2-CSP are not tight under ETH. Moreover, it shows that substantially stronger lower bounds for polynomial-target degeneracy are unlikely to follow from standard reductions from sparse 3-SAT.

*Neapolis University Pafos. Email: braulov2004@gmail.com
†JetBrains Research. Email: buyolitsez1951@gmail.com
‡JetBrains Research. Email: alexander.s.kulikov@gmail.com
§JetBrains Research. Email: ivmihajlin@gmail.com

ISSN 1433-8092

Contents

1 Graph Homomorphism and Target Parameters

A graph homomorphism $G \to H$ is a map from the vertices of a source graph G to the vertices of a target graph H that preserves edges. Homomorphisms to a fixed target encode assignments of local states, while homomorphisms from a small pattern encode global witnesses such as cliques or subgraph counts [Lov12, HN26]. The corresponding decision problem HOM, Graph Homomorphism Problem, asks whether there exists a homomorphism between two given graphs. Various NP-hard graph problems are natural special cases of HOM: for example, G has a proper k-coloring if and only if there exists a homomorphism $G \to K_k$, whereas G contains a k-clique if and only if there is a homomorphism $K_k \to G$. On the other hand, HOM with an n-vertex source graph G and an h-vertex target graph H is a special case of 2-Constraint Satisfaction Problem (2-CSP) with n variables over a domain of size h. Thus, HOM may be viewed as CSP over a graph template.

A straightforward brute-force algorithm (that goes through all mappings) for HOM and 2-CSP has running time $O(h^n) = 2^{O(n \log h)}$. Traxler [Tra08] showed that, for 2-CSP, this upper bound cannot be improved substantially: under the Exponential Time Hypothesis (ETH), there is no algorithm solving 2-CSP in time $h^{o(n)} = 2^{o(n \log h)}$. At the same time, a naive upper bound for the h-Coloring problem (asking whether a given graph admits a proper h-coloring) is also $O(h^n) = 2^{O(n \log h)}$, though it can be solved in single-exponential time: the first such algorithm has running time $O(2.45^n)$ and is due to Lawler [Law76], whereas the best currently known upper bound is $O^*(2^n)$ due to Björklund, Husfeldt, and Koivisto [BHK09] ($O^*(\cdot)$ suppresses multiplicative factors that grow polynomially). Obtaining a similar single-exponential upper bound for HOM would be challenging. As proved by [CHKX06], under ETH, for any $\varepsilon > 0$ and any $k = O(n^{1-\varepsilon})$, there is no algorithm solving k-Clique for n-vertex graphs in time $n^{o(k)}$. Since k-Clique is essentially $\mathsf{HOM}(K_k, \cdot)$, this rules out (under ETH) an upper bound $2^{o(n \log h)}$ in the regime where n is much smaller than h. [CFG+17] proved the same lower bound for *any* regime: under ETH, for any $h = h(n)$, there is no algorithm solving HOM in time $2^{o(n \log h)}$. This holds for the general case when both graphs G and H are part of the input and are not restricted in any way. At the same time, in various applications of Graph Homomorphism, one or both of these graphs may come from a restricted graph family or even be fixed. A broader discussion of homomorphism complexity dichotomies is given in Appendix C.

In recent years, less restrictive graph parameters p have been identified that allow one to solve HOM in time[1] $p(H)^{O(n)}$: treewidth [FHK07], clique-width [Wah11], bounded complement bandwidth [Rzą14], maximum degree [FGKM15], extended clique-width [BD20], track number and persistent majority number [Car26]. These algorithms are faster than $O(h^n)$ and allow one to solve HOM in plain-exponential time $2^{O(n)}$ in the special case when $p(H)$ is bounded. On the other hand, as proved by [FGKM15], chromatic number is too small: under ETH, there is no $\chi(H)^{O(n)}$ algorithm. The known lower and upper bounds with respect to various target graph parameters as well as connections between them are summarized in Figure 1. It is interesting to note that, for each target graph parameter p from the left part of the figure, there is no $p(H)^{o(n)}$ algorithm, under ETH. This follows from the general lower bound due to [CFG+17] via padding (see Observation B.3). Also, the argument of [CFG+17] together with [FGKM15] can be used to prove an optimal ETH-based lower bound $n^{\Omega(n)}$ for graphs H with $\chi(H) = O(1)$ (see Appendix A).

[1] Technically, some of the upper bounds mentioned above have an additional multiplicative term 2^h that is needed to compute a certificate for the corresponding parameter of H (such as coloring or decomposition). When $h = O(n)$ (as in many interesting applications of HOM), this term is absorbed by $p(H)^{O(n)}$. For this reason and to make the summary cleaner, we omit the 2^h term.

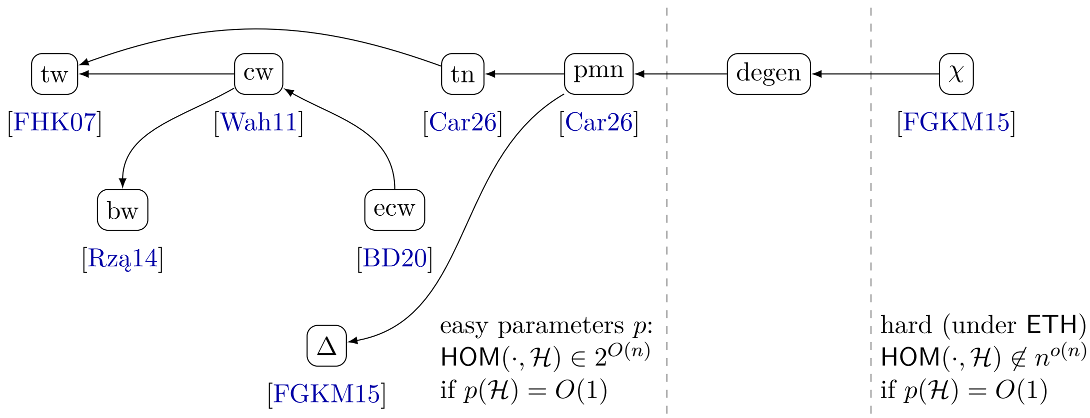

Figure 1: Summary of previously studied target graph parameters and their influence on the complexity of HOM. An arrow from p to q is read as follows: bounded q bounds p; more formally, there exists a function f such that, for any graph H, $p(H) \le f(q(H))$ (all shown connections between parameters are proven in Appendix B). In turn, this implies that an algorithm solving HOM in plain-exponential time in the special case when p is bounded also has plain-exponential time when q is bounded.

1.1 Our Results

We study the complexity of HOM in terms of target *degeneracy* (the minimum integer d such that every nonempty subgraph has a vertex of degree at most d), which is perhaps the most natural unresolved candidate between the known algorithmic and hardness regimes: on the one hand, each of bounded treewidth, bounded maximum degree, and bounded track number implies bounded degeneracy; on the other hand, bounded degeneracy implies bounded chromatic number.

Algorithmically, degeneracy has long served as a measure of sparse structure [Akk73, MB83, CN85, CCC06, AG09, PRS09, ELS10, PRS12], and it appears in tight kernelization bounds for sparse graph problems [CPPW12, DM12, HW12, CGH17]. The HOM counting problem was studied in degenerate graphs, but the degeneracy constraint is usually placed on the large input graph into which a fixed pattern is mapped. Starting with Bressan [Bre21], this line gives near-linear and fine-grained classifications for counting subgraphs, induced subgraphs, homomorphisms, and homomorphic cycles in degenerate graphs [BPS20, BPS21, BGL+22, BR21, GLSY23, PS24, PS25, KKP25]. In those results, the pattern is fixed and the running time is measured as a function of the sparse host, whereas in HOM both G and H are part of the input. There are also related results for bounded outdegree graphs [BLR23], bounded-degeneracy hypergraphs [PS26], and recognition by homomorphism counts to 2-degenerate graphs [Dvo10].

Our results show that the influence of degeneracy of H on the complexity of HOM differs significantly from that of the previously studied parameters. We show that, for every nondecreasing unbounded threshold function D satisfying $D(n) = O(n^{1/3})$ and $D(2n) = O(D(n))$, the promise $\mathrm{degen}(H) \le D(n)$ problem admits no $2^{o(D(n)n)}$ algorithm under ETH.

Theorem 1.1 (Informal version of Theorem 3.1)**.** *Under* ETH*, for every nondecreasing unbounded function $D\colon \mathbb{N} \to \mathbb{N}$ satisfying $D(n) = O(n^{1/3})$ and $D(2n) = O(D(n))$, there is no algorithm solving* HOM *on every instance satisfying* $\mathrm{degen}(H) \le D(n)$ *in time*[2] $O^*\left(2^{o(D(n)n)}\right)$*.*

[2] Recall that $O^*(\cdot)$ suppresses factors that grow polynomially in the input length. For HOM, such factors are of the

We then show that the plain-exponential frontier is already crossed between degeneracy 1 and 2. Targets with degeneracy at most 1 are forests and therefore have bounded treewidth, so they are plain-exponential by [FHK07]. In contrast, we construct hard instances of degeneracy 2, provided that the target is allowed to have size quasi-polynomial in n.

Theorem 1.2 (Informal version of Theorem 4.2)**.** *Under* ETH*, no algorithm solves* HOM *in time* $O^*(n^{o(n)})$ *on instances* (G,H) *satisfying* $\mathrm{degen}(H) \le 2$ *and* $h \le 2^{O(\log^2 n)}$*.*

Finally, we formulate a new barrier showing that improving the lower bound $2^{o(\mathrm{degen}(H)n)}$ further to $2^{\omega(\mathrm{degen}(H)n)}$ would be challenging. Informally, the reason is that such a reduction would have to be stronger than all known fine-grained reductions from 3-SAT. We make it formal below.

By the Sparsification Lemma [IPZ01], ETH can be viewed as hardness of sparse 3-SAT with n variables and $O(n)$ clauses. An explicit sparse formula has bit length $\Theta(n \log n)$, therefore if s denotes the bit size of the input, then sparse 3-SAT is $2^{\Omega(s/\log s)}$ hard. Many fine-grained reductions are known for various problems; however, to the best of our knowledge, not a single one of them establishes a lower bound better than $2^{\Omega(s/\log s)}$, where s denotes the input bit size.

A concrete example of a problem exhibiting a gap between the known lower and upper bounds is the sparse 2-CSP problem. A brute-force algorithm for the 2-CSP with n variables over a domain of size h has running time $O(h^n)$ and there are no $h^{o(n)}$ algorithms under ETH [Tra08]. However, when the number of constraints is linear, the best known lower bound is only $h^{\Omega(n/\log n)}$ [Mar10, KMPS24]. The question of improving the lower bound to $h^{\Omega(n)}$ was raised in [KMPS24]. We note that, in order to close this gap, the lower bound would have to establish a bound stronger than $2^{\Omega(s/\log s)}$, where s denotes the input size in bits; see Section 5.2.

Hypothesis 1.3 (No-Compression Hypothesis, informal version)**.** *There is no fine-grained reduction establishing a lower bound* $2^{\omega(s/\log s)}$ *under* ETH*, where* s *is the bit size of the input.*

A weaker version of this hypothesis was recently considered by Kulikov and Mihajlin [KM24] who proved that an SETH-based lower bound $2^{\Omega(s)}$ violates SETH. We show that refuting a somewhat stronger version of Hypothesis 1.3 implies new circuit lower bounds: if there is a uniform generator that converts $O(n^{0.99})$ input bits into CNF formulas over n variables that are $2^{\Omega(n)}$-hard under ETH, then P^{NP} has truly superlinear circuit size.

Finally, we show that a reduction showing a $2^{\omega(\mathrm{degen}(H)n)}$ lower bound for HOM under ETH would refute Hypothesis 1.3.

Theorem 1.4 (Informal version of Theorem 5.5)**.** *Under Hypothesis 1.3, no reduction can establish a* $2^{\omega(\mathrm{degen}(H)n)}$ *lower bound for* HOM *under* ETH*.*

Techniques. The lower bound of Theorem 1.1 is a reduction from the 3-Coloring problem on graphs of maximum degree 4. We partition N vertices of the instance into vertex-buckets of size r, so that the source of the produced HOM instance has only $O(N/r)$ vertices, while the target must supply a state for every coloring of a bucket. The difficulty is to check the constraints *between* buckets without making these states densely connected; we do this by assigning $O(r)$ labels to the buckets via an equitable coloring, so that every target vertex keeps degree $O(r)$ and hence $\mathrm{degen}(H) = O(r)$, while $\mathrm{degen}(H) \cdot n$ stays linear in N. The list constraints of the intermediate list-homomorphism instance are then removed by attaching a small rigid frame to both graphs. The main difference

form $(n+h)^{O(1)}$.

between Theorem 1.1 and the previous lower bounds for HOM [FGKM15, CFG+17] is that we bucket not only the vertices but also the constraints between them. This approach allows us to keep the target degenerate whereas making the testing problem more difficult.

For Theorem 1.2, we start from the ETH-hardness of sparse bounded-alphabet 2-CSP [KMPS24], encode it into HOM while keeping *both* graphs sparse, and then subdivide every edge into a path of length three; subdivision preserves homomorphisms while making the target 2-degenerate, and sparsity keeps the size growth quasi-polynomial.

The barrier of Theorem 1.4 rests on the following idea: in a D-degenerate target, the set of images still available to a source vertex under some partial homomorphism always admits a record of $O(D \log n)$ bits. This reduces HOM to a succinct homomorphism problem whose instances can be encoded using a few bits, so a reduction proving a $2^{\omega(\text{degen}(H)n)}$ lower bound would compress sparse 3-SAT beyond the $s/\log s$ scale, contradicting Hypothesis 1.3.

Organization. Section 2 fixes notation and assumptions. Section 3 proves the linear degeneracy lower bound. Section 4 proves the constant-degeneracy quasi-polynomial-target lower bound. Section 5 formalizes the no-compression barrier and proves its HOM and sparse CSP consequences. Section 6 proves the circuit lower-bound consequence. Section 7 summarizes the resulting target-side picture and lists open problems. Appendix A gives the polynomial-target bounded-chromatic lower bound.

2 Preliminaries

All logarithms in this paper are taken base 2. For a positive integer q, we write $[q] = \{1, \dots, q\}$. For a nonnegative integer q, by $\text{bin}(q)$ we mean a binary encoding of q. For a partial function f, we write $\text{dom}(f)$ for its domain. All graphs are finite, undirected, and loopless unless explicitly stated otherwise. For a graph G, we write $V(G)$ and $E(G)$ for its vertex and edge sets, $N_G(v)$ for the open neighborhood of v, $\Delta(G)$ for its maximum degree, and $\overline{G}$ for its complement. For a set $S \subseteq V(G)$, the graph $G[S]$ denotes the subgraph induced by S.

Throughout the paper, G denotes the source graph and H denotes the target graph in a homomorphism instance. We write $n = |V(G)|$ and $h = |V(H)|$. In reductions from a source problem, N denotes the size parameter of the original instance.

A hypergraph $\mathcal{A} = (C, \mathcal{E})$ consists of a finite vertex set C and a family $\mathcal{E}$ of subsets of C, called hyperedges. It is d-uniform if every hyperedge has size exactly d. An automorphism of $\mathcal{A}$ is a permutation of C that preserves the hyperedge family, and $\text{Aut}(\mathcal{A})$ denotes the automorphism group.

2.1 Homomorphism Problems

A homomorphism from a graph G to a graph H is a map $\phi\colon V(G) \to V(H)$ such that $uv \in E(G)$ implies $\phi(u)\phi(v) \in E(H)$. We write $G \to H$ if such a map exists. The two-input decision problem is denoted by HOM. For graph classes $\mathcal{G}$ and $\mathcal{H}$, the notation $\text{HOM}(\mathcal{G}, \mathcal{H})$ denotes the restriction to instances (G, H) with $G \in \mathcal{G}$ and $H \in \mathcal{H}$. The symbol "$\cdot$" means that the corresponding side is unrestricted; thus $\text{HOM}(\cdot, H)$ is the fixed-target H-coloring problem and $\text{HOM}(\cdot, \mathcal{H})$ is the restricted variable-target problem with targets from $\mathcal{H}$.

In the List Graph Homomorphism problem, LHOM, the input is a pair of graphs (G, H) together with a list $L(v) \subseteq V(H)$ for every $v \in V(G)$. The goal is to check whether there is a homomorphism $\phi \colon G \to H$ satisfying $\phi(v) \in L(v)$ for every source vertex v.

2.2 Graph Parameters

We use the following basic graph parameters.

- The chromatic number $\chi(G)$ is the least number of colors in a proper vertex coloring of G.
- The maximum degree $\Delta(G)$ is the maximum degree of a vertex of G.
- The degeneracy $\mathrm{degen}(G)$ is the least integer d such that every non-empty subgraph of G has a vertex of degree at most d. Equivalently, G has an ordering in which each vertex has at most d later neighbors.

The remaining target-side parameters appearing in Figure 1 are defined formally in Appendix B, where we also state parameter implications.

2.3 Constraint Satisfaction

A finite-domain CSP instance consists of variables, a finite domain, and constraints restricting tuples of variables. In this paper, the source problems are binary CSPs. For $h \geq 2$, a $(2, h)$-CSP instance has variables taking values in $[h]$, unary constraints $R_x \subseteq [h]$, and binary constraints $R_{xy} \subseteq [h]^2$ on ordered pairs of distinct variables. An assignment satisfies the instance if it satisfies every unary and binary relation. The constraint graph has one vertex for each variable and one edge for each binary constraint, with no orientation. More generally, a (k, h)-CSP instance has n variables over $[h]$ and constraints of arity at most k. A *sparse* (k, h)-CSP instance is one with at most $O(n)$ constraints. In the encoding of CSP, every constraint encodes its scope and the truth table of its relation.

2.4 Boolean Circuits

A Boolean circuit C over variables $x_1, \dots, x_n$ is a directed acyclic graph with nodes of in-degree zero or two. The in-degree zero nodes are labeled by variables x_i and constants 0 or 1, whereas the in-degree two nodes are labeled by binary Boolean functions. The only out-degree zero node is the output of the circuit. A circuit C computes a Boolean function $f \colon \{0,1\}^n \to \{0,1\}$ in a natural way.

For a language $L \subseteq \{0,1\}^*$, a circuit C with n input variables computes L on inputs of length n if, for every $x \in \{0,1\}^n$, $C(x) = 1$ exactly when $x \in L$. For a language L, let $\mathsf{CC}_L[n]$ denote the minimum size of a Boolean circuit computing L on all inputs of length n. For a function $s \colon \mathbb{N} \to \mathbb{N}$, write

$$L \in \mathsf{SIZE}[s(n)]$$

if for every sufficiently large n,

$$\mathsf{CC}_L[n] \leq s(n).$$

Proving lower bounds on the circuit size of explicit functions is a long-standing challenge. A counting argument due to Shannon [Sha49] shows that almost all Boolean functions on n variables require circuits of size $\Omega(2^n/n)$. However, the best known lower bound for an explicit function[3] is $3.1n - o(n)$, due to Li and Yang [LY22].

For a string $a \in \{0,1\}^N$, let $\mathsf{CC}[a]$ denote the minimum size of a circuit on $k = \lceil \log_2 N \rceil$ input bits whose first N truth-table values are the bits of a. The remaining $2^k - N$ truth-table values are ignored.

2.5 SAT Complexity Hypotheses

A k-CNF formula is a conjunction of clauses of width at most k, and k-SAT asks whether such a formula is satisfiable. A sparse 3-SAT instance on N variables has $O(N)$ clauses.

Hypothesis 2.1 (Exponential Time Hypothesis, ETH [IPZ01])**.** *There is no algorithm solving* 3-SAT *on formulas with N variables in time $2^{o(N)}$.*

By the Sparsification Lemma of Impagliazzo, Paturi, and Zane [IPZ01], ETH is equivalent to the statement that sparse 3-SAT has no $2^{o(N)}$-time algorithm.

Hypothesis 2.2 (Strong Exponential Time Hypothesis, SETH [IP01])**.** *For every $\varepsilon > 0$, there exists an integer $k \geq 3$ such that k-SAT on formulas with N variables cannot be solved in time $2^{(1-\varepsilon)N} \operatorname{poly}(N)$.*

Hypothesis 2.3 (Counting Strong Exponential Time Hypothesis, $\#\mathsf{SETH}$ [DHM+14, CM16])**.** *For every $\varepsilon > 0$, there exists an integer $k \geq 3$ such that counting the satisfying assignments of k-CNF formulas with N variables cannot be done in time $2^{(1-\varepsilon)N} \operatorname{poly}(N)$.*

The following bounded-degree coloring consequence of ETH is due to Kochol, Lozin, and Randerath [KLR03].

Theorem 2.4. *Assuming* ETH*, there exists a constant $\delta > 0$ such that* 3-$\mathsf{Coloring}$ *on N-vertex graphs of maximum degree at most 4 cannot be solved in time $2^{\delta N}$.*

Throughout the paper, we use the notion of a fine-grained reduction; for its formal definition and an overview of results in fine-grained complexity, see [Wil18].

3 Linear Dependence on Degeneracy

This section reduces 3-$\mathsf{Coloring}$ on bounded-degree graphs to HOM, on targets whose degeneracy is small. The proof has two separate parts. First, we reduce 3-$\mathsf{Coloring}$ to an LHOM instance whose target already has small degeneracy. Then, we encode the list constraints into ordinary graph homomorphism. The construction is arranged so that the target degeneracy, multiplied by the number of source vertices, stays linear in the size of the original instance. Consequently, an algorithm whose running-time exponent were sublinear in this product would solve bounded-degree 3-$\mathsf{Coloring}$ faster than ETH permits.

Theorem 3.1 (Formal version of Theorem 1.1)**.** *Unless* ETH *fails, for every nondecreasing unbounded function $D\colon \mathbb{N} \to \mathbb{N}$ satisfying $D(n) = O(n^{1/3})$ and $D(2n) = O(D(n))$, there is no algorithm that solves* HOM *on every instance (G,H) satisfying $\operatorname{degen}(H) \leq D(n)$ in time*

$$O^*\left(2^{o(D(n)n)}\right).$$

[3] By an explicit function, we mean a function computable in P^{NP}.

More precisely, for every sufficiently large integer r and all sufficiently large N, every N-vertex graph Q of maximum degree at most 4 can be transformed into an instance (G_r, H_r) of HOM*, in time polynomial in $|Q| + |H_r|$, with*

$$|V(G_r)| = O(N/r + r^2), \qquad |V(H_r)| = 2^{O(r^2)},$$

and with

$$\mathrm{degen}(H_r) = O(r),$$

such that Q is 3-colorable if and only if there is a homomorphism $G_r \to H_r$.

3.1 Proof Overview

We take a graph Q on N vertices with maximum degree at most 4, and partition its vertices into $\lceil N/r \rceil$ *vertex-buckets* of size at most r; we fold each vertex-bucket into a single vertex of a list-homomorphism source.

The obstacle is that folding r vertices into one source vertex forces the target to supply, for each group, a vertex for every possible color pattern in $[3]^r$. The target may therefore have $2^{O(r^2)}$ vertices, and the whole difficulty is to verify the 3-coloring constraints *between* groups without making these vertices densely connected. Classical reductions do exactly that: they attach to each state a consistency gadget that lists all compatible states of its neighbors, and since a single group can interact with $\Theta(r)$ others, the resulting states have degree exponential in r.

We avoid this by further partitioning $E(Q)$ into *edge-buckets* of at most r edges each. The partition is chosen so that the vertex-buckets meeting in one edge-bucket have distinct *labels*; hence one target state can refer to them by label without ambiguity.

The list target has two kinds of state vertices, plus a test board. A *bucket-state* vertex stores one color string $c \in [3]^r$ for one labeled bucket. An *edge-state* vertex stores only the local data for one edge-bucket: the color strings of the participating labels and, for each edge in that edge-bucket, the two bucket positions that it connects. The global *test board* contains small test vertices that check one edge constraint at a time.

The point of edge splitting is locality in checking. Since an edge-state vertex is adjacent only to the bucket-states of its own participating labels and to the test vertices for its own edge-bucket, its degree is $O(r)$ even though there may be $2^{O(r^2)}$ edge-states in total.

After constructing this sparse list-homomorphism instance, we remove the lists. We add a small rigid *frame* to both graphs and connect each source vertex to the frame according to the target vertices it is allowed to use. Rigidity forces every homomorphism to respect these connections, so the lists are simulated inside ordinary HOM.

The rest of the section follows this order. Section 3.2 builds the buckets and labels. Section 3.3 defines the LHOM instance and proves that it is equivalent to 3-Coloring. Section 3.4 replaces lists by the rigid frame. Section 3.5 finishes the parameter bounds and derives the ETH contradiction.

3.2 Bucket Decomposition

Let Q be an N-vertex graph of maximum degree at most 4. Fix an integer $r \geq 2$. We construct an LHOM instance $(\Lambda_r, \Gamma_r, \mathcal{L}_r)$, where Λ_r is the source graph, Γ_r is the target graph, and $\mathcal{L}_r$ is the list assignment. Thus, for each source vertex $v \in V(\Lambda_r)$, the set $\mathcal{L}_r(v) \subseteq V(\Gamma_r)$ consists of the allowed images of v. Later, we convert it into an ordinary homomorphism instance (G_r, H_r).

The reduction groups the vertices and edges of Q into small blocks. A *vertex-bucket* is a set of at most r vertices of Q; one source vertex of Λ_r will be responsible for the colors of an entire vertex-bucket. An *edge-bucket* is a set of at most r edges of Q; one source vertex of Λ_r will be responsible for checking all the 3-coloring constraints inside one edge-bucket. We say that a vertex-bucket *participates* in an edge-bucket if it contains an endpoint of some edge of that edge-bucket.

A vertex of Λ_r that handles an edge-bucket must know the buckets whose colors it compares, so we give every vertex-bucket a *label* from a small set $[L]$ with $L = O(r)$, and we arrange that the buckets participating in any single edge-bucket carry pairwise distinct labels. The next lemma produces such a bucketing and labeling.

Lemma 3.2. *For any integer $r \geq 2$ and all sufficiently large N, any graph Q on N vertices with $\Delta(Q) \leq 4$ admits:*

- *a partition of $V(Q)$ into vertex-buckets $B_1, \ldots, B_k$ of size at most r;*
- *a label map $\lambda\colon \{B_1, \ldots, B_k\} \to [L]$ with $L \coloneqq 4r + 1$;*
- *a partition of $E(Q)$ into edge-buckets $E_1, \ldots, E_s$ with $s = O(N/r + r)$ and $|E_j| \leq r$,*

such that, inside each edge-bucket, no label occurs on two different participating vertex-buckets. Moreover, finding this partition and labeling can be done in polynomial time.

Proof. Partition $V(Q)$ arbitrarily into buckets of size at most r, and let $k = \lceil N/r \rceil$. Let $\mathcal{B}$ be the bucket graph: two buckets are adjacent if some edge of Q has one endpoint in each. Since each bucket has at most r vertices and Q has maximum degree at most 4, we have $\Delta(\mathcal{B}) \leq 4r$. By the Hajnal–Szemerédi equitable coloring theorem [HS70], $\mathcal{B}$ has an equitable proper coloring with $L = 4r + 1$ colors; denote it by $\lambda\colon \{B_1, \ldots, B_k\} \to [L]$.[4] Each label is used by $O(k/L + 1) = O(N/r^2 + 1)$ buckets. Note that this coloring can be found in polynomial time, see e.g. [KKMS10].

Create a multigraph M on vertex set $[L]$. Each edge of Q whose endpoint buckets have labels a and b contributes an edge ab to M; an edge internal to one bucket contributes a loop. Since λ is proper on $\mathcal{B}$, loops come only from edges internal to a bucket. For a fixed label a, the total number of edges incident with buckets of label a is at most $4r \cdot O(N/r^2 + 1) = O(N/r + r)$. The line graph of this multigraph has maximum degree $O(N/r + r)$. A greedy coloring of this line graph therefore splits the edges of Q into $O(N/r + r)$ classes, where each class is a matching in M.

These matching classes may still contain more than r edges, so we split every class arbitrarily into chunks of size at most r and declare the chunks to be the edge-buckets. If the matching classes are $F_1, \ldots, F_t$, where $t = O(N/r + r)$, then the number of chunks is at most

$$\sum_{i=1}^{t} \lceil |F_i|/r \rceil \leq |E(Q)|/r + t = O(N/r) + O(N/r + r),$$

since $|E(Q)| \leq 2N$. Thus the total number of edge-buckets is $O(N/r + r)$, and each edge-bucket contains at most r edges.

It remains to check the distinct-label property. Fix an edge-bucket $E_j \subseteq E(Q)$. By construction, all edges in E_j came from one matching class in M. For an edge $e \in E_j$, let $\mu(e)$ be the edge of M produced by the labels of the endpoint buckets of e. If e is internal to a single vertex-bucket B,

[4] An equitable coloring is a proper coloring such that the sizes of any two color classes differ by at most one.

then $\mu(e)$ is a loop at $\lambda(B)$ and e contributes only the participating bucket B. If e has endpoints in two different buckets B and B', then B and B' are adjacent in $\mathcal{B}$; since λ is a proper coloring of $\mathcal{B}$, we have $\lambda(B) \neq \lambda(B')$.

Now take two distinct edges $e, f \in E_j$. The edges $\mu(e)$ and $\mu(f)$ are distinct edges of one matching in M, so they have no common incident label. Therefore no label used by a vertex-bucket participating in e can also be used by a vertex-bucket participating in f. So, the labels on all vertex-buckets participating in E_j are pairwise distinct. □

For the rest of the construction, we pad each vertex-bucket B_i with dummy vertices to make it of size exactly r.

3.3 List-Homomorphism Instance

The target. Fix a bucketing and labeling from Lemma 3.2 for the rest of the section. We define the list target graph Γ_r. Its vertices fall into three types. A *bucket-state* vertex stores one color string for one labeled vertex-bucket. An *edge-state* vertex stores the color strings of the labels participating in one edge-bucket, together with a partial map recording which two bucket-coordinates each edge compares. The *test board* is a family of test vertices; each one checks a declared endpoint of an edge, or rules out one monochromatic edge.

The bucket-state set is

$$S_B = \{U_{\ell,c} \colon \ell \in [L],\ c \in [3]^r\}.$$

A bucket-state represents a coloring string for one bucket of label ℓ.

The edge-state set is

$$S_E := \left\{ W_{Z,c,\Pi} \colon \begin{array}{c} \varnothing \neq Z \subseteq [L],\ |Z| \leq 2r, \\ c = (c_\ell)_{\ell \in Z},\ c_\ell \in [3]^r \text{ for every } \ell \in Z, \\ \Pi \colon \operatorname{dom}(\Pi) \subseteq [r] \to Z \times [r] \times Z \times [r] \end{array} \right\}.$$

That is, Π is a partial function from $[r]$ to $Z \times [r] \times Z \times [r]$. If $\Pi(p) = (a, x, b, y)$, then the edge numbered p compares coordinate x of label a with coordinate y of label b; the test board, defined next, verifies this claim and the resulting color condition. We make Π a partial function so that an edge-state compares only the edges of its own edge-bucket.

The test board is the union of the following families:

$$\begin{aligned} T_A &= \{A_{p,a} \colon p \in [r],\ a \in [L]\}, & T_B &= \{B_{p,b} \colon p \in [r],\ b \in [L]\}, \\ T_X &= \{X_{p,x} \colon p, x \in [r]\}, & T_Y &= \{Y_{p,y} \colon p, y \in [r]\}, \\ T_C &= \{C_{p,\gamma} \colon p \in [r],\ \gamma \in [3]\}. \end{aligned}$$

Let $T = T_A \cup T_B \cup T_X \cup T_Y \cup T_C$.

The list target graph has vertex set

$$V(\Gamma_r) = S_B \sqcup S_E \sqcup T.$$

Its edges are defined as follows.

- State edges. For $U_{\ell,c} \in S_B$ and $W_{Z,d,\Pi} \in S_E$, add the edge

$$U_{\ell,c} W_{Z,d,\Pi}$$

if and only if

$$\ell \in Z \quad \text{and} \quad d_\ell = c.$$

- Test edges. For $W = W_{Z,c,\Pi} \in S_E$ and $\Pi(p) = (a, x, b, y)$, add the metadata edges

  $$WA_{p,a}, \qquad WX_{p,x}, \qquad WB_{p,b}, \qquad WY_{p,y}.$$

  For $\gamma \in [3]$, add the color-check edge $WC_{p,\gamma}$ if and only if

  $$\Pi(p) = (a, x, b, y) \quad \text{and} \quad \neg(c_a[x] = \gamma = c_b[y]).$$

  The expression is evaluated only when $\Pi(p)$ is defined, which already implies $a, b \in Z$. If $a = b$, the same string c_a is used in both positions. See Figure 2.

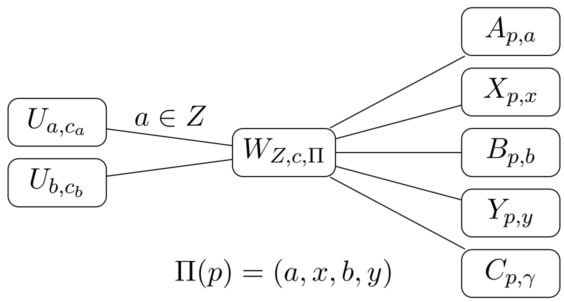

Figure 2: One edge-state stores only the tests used by its edge-bucket. The test forces the tuple $\Pi(p)$, and the color-check vertex $C_{p,\gamma}$ forbids the monochromatic case.

Lemma 3.3.

$$|V(\Gamma_r)| = 2^{O(r^2)}.$$

Proof. We have

$$|S_B| = L3^r = \Theta(r\,3^r)$$

and

$$|T| = 2Lr + 2r^2 + 3r = O(r^2).$$

For the edge-states, for $\mu = |Z| \le 2r$, first choose Z, then choose μ color strings in $[3]^r$, choose a domain for Π, and finally choose a value in $Z \times [r] \times Z \times [r]$ for each edge index in that domain. Hence

$$|S_E| \le \sum_{\mu=1}^{2r} \binom{L}{\mu} (3^r)^\mu \sum_{q=0}^{r} \binom{r}{q} (\mu^2 r^2)^q.$$

Since $L = \Theta(r)$ and $\mu \le 2r$, the logarithm of this expression is $O(r^2)$: the color-string factor contributes $O(r^2)$ and the remaining combinatorial factors contribute only $O(r \log r)$. Thus $|S_E| \le 2^{O(r^2)}$, and therefore $|V(\Gamma_r)| = 2^{O(r^2)}$. ☐

The source. The source graph records the bucket structure of Q: it has a bucket vertex for each vertex-bucket, to be sent to a bucket-state, and an edge-bucket vertex for each edge-bucket, to be sent to an edge-state. A single copy of the test board is shared by all edge-buckets.

The graph Λ_r contains:

- a copy T' of the entire test board T;
- one bucket vertex u_i for each vertex-bucket B_i;
- one edge-bucket vertex w_j for each edge-bucket E_j.

Add the following edges. First, join w_j to u_i if and only if B_i participates in E_j. Second, index the edges in each edge-bucket by distinct values $p \in [|E_j|] \subseteq [r]$ and orient them arbitrarily. If the edge with index p is oriented from coordinate x in bucket B_α to coordinate y in bucket B_β, put $a = \lambda(B_\alpha)$ and $b = \lambda(B_\beta)$, and add the following edges:

$$w_j A'_{p,a}, \quad w_j X'_{p,x}, \quad w_j B'_{p,b}, \quad w_j Y'_{p,y},$$

and the three edges $w_j C'_{p,\gamma}$, for $\gamma \in [3]$. The same global copied test vertex may be used by many edge-buckets.

Lemma 3.4.

$$|V(\Lambda_r)| = O(N/r + r^2).$$

Proof. There are $\lceil N/r \rceil$ bucket vertices and $O(N/r + r)$ edge-bucket vertices by Lemma 3.2. The copied test board has $O(r^2)$ vertices. □

Markers and lists. We now introduce markers, which are auxiliary names used only to define list constraints. Each source and target vertex receives a set of markers. A source vertex v may be mapped precisely to those target vertices x for which $A_\Lambda(v) \subseteq A_\Gamma(x)$, where A_Λ and A_Γ denote the marker sets associated with the vertices of Λ and Γ, respectively. Throughout this construction, we write $\beta_L := \lceil \log_2 L \rceil$ and $\beta_r := \lceil \log_2 r \rceil$ for the label- and coordinate-index lengths. The markers are:

- selector markers $\rho_U, \rho_W, \rho_A, \rho_B, \rho_X, \rho_Y, \rho_C$;
- for every label bit $h \in [\beta_L]$ and bit value $b \in \{0, 1\}$, markers $\mathsf{D}_{h,b}$, $\mathsf{A}_{h,b}$, and $\mathsf{B}_{h,b}$;
- for every coordinate bit $h \in [\beta_r]$ and bit value $b \in \{0, 1\}$, markers $\mathsf{X}_{h,b}$ and $\mathsf{Y}_{h,b}$;
- for every edge-index bit $h \in [\beta_r]$ and bit value $b \in \{0, 1\}$, a marker $\mathsf{P}_{h,b}$;
- color markers $\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3$.

Thus the number of markers is

$$M := 7 + 6\beta_L + 6\beta_r + 3 = O(\log r).$$

For $m \in \{L, r\}$ and $z \in [m]$, let $\mathrm{bin}_m(z)$ be the fixed-length binary encoding of $z-1 \in \{0, \dots, m-1\}$ with $\lceil \log_2 m \rceil$ bits. For a label $\ell \in [L]$, define

$$\begin{aligned}
\mathsf{D}(\ell) &:= \{\mathsf{D}_{h,\mathrm{bin}_L(\ell)_h} : h \in [\beta_L]\},\\
\mathsf{A}(\ell) &:= \{\mathsf{A}_{h,\mathrm{bin}_L(\ell)_h} : h \in [\beta_L]\},\\
\mathsf{B}(\ell) &:= \{\mathsf{B}_{h,\mathrm{bin}_L(\ell)_h} : h \in [\beta_L]\}.
\end{aligned}$$

For a coordinate $x \in [r]$, define

$$\mathsf{X}(x) \coloneqq \{\mathsf{X}_{h,\mathrm{bin}_r(x)_h} : h \in [\beta_r]\},$$
$$\mathsf{Y}(x) \coloneqq \{\mathsf{Y}_{h,\mathrm{bin}_r(x)_h} : h \in [\beta_r]\}.$$

For an edge index $p \in [r]$, define

$$\mathsf{Pcode}(p) \coloneqq \{\mathsf{P}_{h,\mathrm{bin}_r(p)_h} : h \in [\beta_r]\}.$$

Target marker sets. The target marker sets are

$$A_\Gamma(U_{\ell,c}) = \{\rho_U\} \cup \mathsf{D}(\ell),$$

$$A_\Gamma(W_{Z,c,\Pi}) = \{\rho_W\},$$

and, for test vertices,

$$A_\Gamma(A_{p,a}) = \{\rho_A\} \cup \mathsf{Pcode}(p) \cup \mathsf{A}(a), \qquad A_\Gamma(B_{p,b}) = \{\rho_B\} \cup \mathsf{Pcode}(p) \cup \mathsf{B}(b),$$

$$A_\Gamma(X_{p,x}) = \{\rho_X\} \cup \mathsf{Pcode}(p) \cup \mathsf{X}(x), \qquad A_\Gamma(Y_{p,y}) = \{\rho_Y\} \cup \mathsf{Pcode}(p) \cup \mathsf{Y}(y),$$
$$A_\Gamma(C_{p,\gamma}) = \{\rho_C, \mathsf{C}_\gamma\} \cup \mathsf{Pcode}(p).$$

Source marker sets. The source marker sets are

$$A_\Lambda(u_i) = \{\rho_U\} \cup \mathsf{D}(\lambda(B_i)),$$
$$A_\Lambda(w_j) = \{\rho_W\},$$

and each copied test vertex has the marker set of its target namesake, for example, $A_\Lambda(A'_{p,a}) = \{\rho_A\} \cup \mathsf{Pcode}(p) \cup \mathsf{A}(a)$, and the same applies to $B'_{p,b}$, $X'_{p,x}$, $Y'_{p,y}$, and $C'_{p,\gamma}$.

The lists are defined by marker containment:

$$\mathcal{L}_r(v) = \{x \in V(\Gamma_r) : A_\Lambda(v) \subseteq A_\Gamma(x)\}.$$

Thus each bucket vertex u_i lists precisely to bucket-states of label $\lambda(B_i)$, each copied test vertex has a singleton list, and each edge-bucket vertex lists to all edge-states.

Correctness. For completeness, a proper 3-coloring of Q is read off into the target: each bucket vertex goes to the bucket-state holding its true color string, and each edge-bucket vertex goes to the edge-state holding those strings and the map for its edges.

Lemma 3.5. *If Q is 3-colorable, then $(\Lambda_r, \Gamma_r, \mathcal{L}_r)$ has a list homomorphism.*

Proof. Let $\kappa \colon V(Q) \to [3]$ be a proper coloring. Map the copied test board T' to the identically named tests in T.

For a bucket B_i, let $c_i \in [3]^r$ be any string satisfying $c_i[v] = \kappa(v)$ for every vertex $v \in B_i$; fill unused coordinates arbitrarily. Set

$$\phi(u_i) = U_{\lambda(B_i),c_i}.$$

For an edge-bucket E_j, let

$$I_j = \{i \colon B_i \text{ participates in } E_j\}$$

and define

$$Z_j = \{\lambda(B_i) \colon i \in I_j\}.$$

The labels in Z_j are distinct by Lemma 3.2. Define strings d_ℓ for $\ell \in Z_j$ by $d_{\lambda(B_i)} = c_i$ for $i \in I_j$. This is well-defined because the labels $\lambda(B_i)$, $i \in I_j$, are pairwise distinct.

Define a partial function Π_j as follows. If the edge with index p in E_j is oriented from coordinate x in bucket B_α to coordinate y in bucket B_β, then set

$$\Pi_j(p) = (\lambda(B_\alpha), x, \lambda(B_\beta), y).$$

Let $d = (d_\ell)_{\ell \in Z_j}$. Set

$$\phi(w_j) = W_{Z_j, d, \Pi_j}.$$

The assigned images lie in the required lists by the marker definitions. If $w_j u_i$ is an edge of Λ_r, then B_i participates in E_j, so the chosen edge-state explicitly contains the bucket-state assigned to u_i; hence the state edge is present in Γ_r.

Finally consider the test edges from w_j for the edge with index p, oriented from coordinate x in bucket B_α to coordinate y in bucket B_β. Put $a = \lambda(B_\alpha)$ and $b = \lambda(B_\beta)$. By construction, $\Pi_j(p) = (a, x, b, y)$, so the metadata edges to $A_{p,a}$, $X_{p,x}$, $B_{p,b}$, and $Y_{p,y}$ are present in Γ_r. Since κ is proper, the two endpoint colors are not both γ for any $\gamma \in [3]$, and therefore all color-check edges to $C_{p,\gamma}$ are present. $\square$

For soundness, the lists force every source vertex into its intended type, the state edges make neighboring buckets agree on their shared color strings, and a missing color-check edge would witness a monochromatic edge of Q, which a list homomorphism cannot produce.

Lemma 3.6. *If $(\Lambda_r, \Gamma_r, \mathcal{L}_r)$ has a list homomorphism, then Q is 3-colorable.*

Proof. Let $\phi \colon \Lambda_r \to \Gamma_r$ be a list homomorphism.

Because $\phi(u_i) \in \mathcal{L}_r(u_i)$, the marker definitions imply that $\phi(u_i)$ is a bucket-state with label $\lambda(B_i)$. Thus

$$\phi(u_i) = U_{\lambda(B_i), c_i}$$

for some $c_i \in [3]^r$.

Similarly, $\phi(w_j) \in \mathcal{L}_r(w_j)$ implies

$$\phi(w_j) = W_{Z_j, d, \Pi_j}$$

for some edge-state. Since $w_j u_i$ is an edge whenever B_i participates in E_j, the definition of state edges forces

$$\lambda(B_i) \in Z_j \quad \text{and} \quad d_{\lambda(B_i)} = c_i \qquad \text{for every participating } B_i.$$

Each copied test vertex has a singleton list determined by its selector and binary code. For example, $\mathcal{L}_r(A'_{p,a}) = \{A_{p,a}\}$: the selector chooses the A-family, and the two binary codes determine the edge index and label. Thus $A'_{p,a}$ maps to $A_{p,a}$, and the same argument applies to the B, X, Y, and C tests.

The strings c_i define a coloring of all real vertices of Q by assigning each $v \in B_i$ the color $c_i[v]$. Suppose some edge is monochromatic. Let this edge lie in E_j, and suppose, with the orientation used in the construction, that its endpoints are position x in B_α and position y in B_β. Put

$$\gamma = c_\alpha[x] = c_\beta[y].$$

Let p be the index of this edge. The graph Λ_r contains the metadata edges from w_j to $A'_{p,\lambda(B_\alpha)}$, $X'_{p,x}$, $B'_{p,\lambda(B_\beta)}$, and $Y'_{p,y}$, and it also contains the color-check edge $w_j C'_{p,\gamma}$. The image of w_j is an edge-state whose strings on the labels $\lambda(B_\alpha)$ and $\lambda(B_\beta)$ are exactly c_α and c_β, by the state edges to u_α and u_β. The metadata edges force $\Pi_j(p) = (\lambda(B_\alpha), x, \lambda(B_\beta), y)$. Hence the condition $c_\alpha[x] = \gamma = c_\beta[y]$ causes the corresponding color-check edge to $C_{p,\gamma}$ to be absent in Γ_r. This contradicts that ϕ is a homomorphism. Therefore the induced coloring of Q is proper. $\square$

Corollary 3.7. *For every input graph Q of maximum degree at most 4,*

$$Q \text{ is 3-colorable} \iff (\Lambda_r, \Gamma_r, \mathcal{L}_r) \text{ has a list homomorphism.}$$

3.4 Removing the Lists

A *frame* is a small rigid graph equipped with many pairwise disjoint groups of vertices, the *ports*. Its purpose is to simulate, inside ordinary graph homomorphism, the list constraints that the reduction needs: we attach a source vertex to the ports that encode its intended type, and rigidity guarantees that these incidences survive every homomorphism. Concretely, if a source vertex is complete to a chosen set of ports (that is, adjacent to every vertex of each of these ports), then the image of that vertex under any homomorphism must be complete to the corresponding target ports.

We build the frame in two steps. First we construct a 3-uniform hypergraph that admits no nontrivial automorphism yet contains M disjoint perfect matchings into triples.

Lemma 3.8. *For every $M \geq 1$, there is a 3-uniform hypergraph $\mathcal{A} = (C, \mathcal{E})$ and pairwise edge-disjoint subfamilies $\mathcal{C}_1, \ldots, \mathcal{C}_M \subseteq \mathcal{E}$ such that $|C| = q = O(M)$, $q \geq 6$, $3 \mid q$, every $\mathcal{C}_i$ is a partition of C into triples, $|\mathcal{E}| = O(M^2)$, and $\mathrm{Aut}(\mathcal{A})$ is trivial.*

Proof. Let $N = \max\{M, 10\}$ and put

$$C = C_A \sqcup C_B \sqcup C_C, \text{ where } C_A = \{a_1, \ldots, a_N\}, \quad C_B = \{b_1, \ldots, b_N\}, \quad C_C = \{c_1, \ldots, c_N\}.$$

Thus

$$q = |C| = 3N = O(M), \qquad q \geq 30, \qquad 3 \mid q.$$

For $i \in [M]$, set $r_i = i - 1$ and define

$$\mathcal{C}_i = \big\{\{a_x, b_{x+r_i}, c_{x+2r_i}\} \colon x \in [N]\big\},$$

where subscripts are read modulo N, with representatives in $[N]$.

Each $\mathcal{C}_i$ is a partition of C into triples. The families $\mathcal{C}_1, \ldots, \mathcal{C}_M$ are pairwise edge-disjoint. Indeed, if

$$\{a_x, b_{x+r_i}, c_{x+2r_i}\} = \{a_y, b_{y+r_j}, c_{y+2r_j}\},$$

then the disjointness of C_A, C_B, C_C gives $x = y$ and $r_i \equiv r_j \pmod{N}$. Since $0 \leq r_i, r_j \leq M - 1 \leq N - 1$, we get $r_i = r_j$, and hence $i = j$.

It remains to add marker edges that destroy all automorphisms. Let s be the least integer such that $\binom{s}{2} \geq q$. Since $N \geq 10$, we have $\binom{N}{2} \geq 3N = q$, so $s \leq N$. Set

$$S = \{a_1, \ldots, a_s\} \subseteq C_A, \qquad \ell = |C \setminus S| = q - s.$$

List the pairs of S in lexicographic order as $P_1, P_2, \ldots, P_{\binom{s}{2}}$. Enumerate $C \setminus S$ as $\{v_1, \ldots, v_\ell\}$. For each $j \in [\ell]$, add the marker edges (since $\ell \leq q \leq \binom{s}{2}$, the first ℓ pairs are defined)

$$P_t \cup \{v_j\}, \qquad 1 \leq t \leq j.$$

Let $\mathcal{R}$ be the resulting marker family, and define

$$\mathcal{E} = \mathcal{R} \sqcup \mathcal{C}_1 \sqcup \cdots \sqcup \mathcal{C}_M.$$

We now prove that marker degrees distinguish all vertices. For a vertex $v_j \in C \setminus S$, we have $\deg_{\mathcal{R}}(v_j) = j$, so their degrees are precisely $[\ell]$. For others, write

$$d_i = \deg_{\mathcal{R}}(a_i), \qquad 1 \leq i \leq s.$$

If P_t contains a_i, then this pair contributes $\ell - t + 1$ to d_i, because it appears with precisely the vertices v_j satisfying $j \geq t$.

Since $P_1, \ldots, P_{\binom{s}{2}}$ are in lexicographic order, $d_1 > d_2 > \cdots > d_s$. We also claim that every d_i exceeds ℓ. Since $q \geq 30$, the minimality of s implies $s \geq 9$. If $s = 9$, then $q \geq 30 = 4s - 6$. If $s \geq 10$, then

$$q \geq \binom{s-1}{2} + 1 \geq 4s - 6.$$

Thus $\ell = q - s \geq 3s - 6$. The pairs $\{a_1, a_s\}$, $\{a_2, a_s\}$, $\{a_3, a_s\}$ occur in the lexicographic list at positions $s - 1$, $2s - 3$, and $3s - 6$, respectively. All three are therefore used, and they all contain a_s. Consequently

$$d_s \geq (\ell - (s-1) + 1) + (\ell - (2s-3) + 1) + (\ell - (3s-6) + 1) = 3\ell - 6s + 13 \geq \ell + 1.$$

Hence $d_1 > d_2 > \cdots > d_s > \ell$. All vertices of C therefore have distinct marker degrees.

Every vertex belongs to exactly one edge of each $\mathcal{C}_i$, so every vertex has total degree $M + \deg_{\mathcal{R}}(v)$ in $\mathcal{A} = (C, \mathcal{E})$. The total degrees are distinct, and every automorphism preserves total degree. Thus every automorphism fixes every vertex, and $\operatorname{Aut}(\mathcal{A}) = \{1\}$.

Finally,

$$|\mathcal{E}| = MN + |\mathcal{R}| = MN + \sum_{j=1}^{\ell} j \leq MN + \frac{q(q+1)}{2} = O(M^2).$$

□

Now we turn this hypergraph skeleton into a graph by blowing up each hyperedge into a rigid graph gadget, which yields the frame together with its M ports. The gadget is a copy of the fixed graph J provided by the next lemma.

Lemma 3.9. *There is a connected, triangle-free graph J with $\chi(J) = 4$ that is a core, that is, every endomorphism of J is an automorphism.*

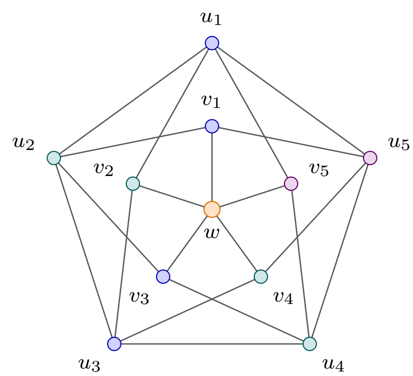

Figure 3: The Grötzsch graph. It is connected, triangle-free, 4-chromatic, and every endomorphism is an automorphism.

Proof. Let J be the Grötzsch graph [Grö59], depicted in Figure 3. It is connected, triangle-free, and satisfies $\chi(J) = 4$. Moreover, every proper subgraph of J is 3-colorable. If an endomorphism were not onto the whole vertex set, its image would be a proper subgraph and hence 3-colorable, giving a 3-coloring of J, a contradiction. If it is onto the whole vertex set, then, since J is finite, it is bijective and therefore an automorphism. □

We turn the construction in Lemma 3.8 into a graph frame. The vertex set C of the hypergraph becomes a clique, whose vertices we call *central*; being the unique maximum clique, it must be preserved by every homomorphism, so a homomorphism induces a permutation σ of C. Each hyperedge is then realized by a graph gadget, which is a copy of a fixed template, attached so that it can survive only if σ maps hyperedges to hyperedges.

Lemma 3.10. *For every integer $M \geq 1$, there is a graph F_M with pairwise disjoint non-empty port sets $P_1, \ldots, P_M \subseteq V(F_M)$ such that the following hold.*

- *$|V(F_M)| = O(M^2)$ and $|P_i| = O(M)$ for every i.*
- *No vertex of F_M is adjacent to every vertex of any port P_i.*
- *Let B be any bipartite graph. Form a graph X from the disjoint union of F_M and B by making each $b \in V(B)$ complete to a union of ports $\bigcup_{i\in A(b)} P_i$, where $A(b) \subseteq [M]$. Then every homomorphism $F_M \to X$ is an automorphism of F_M and maps every port P_i onto itself.*

Proof. Let J be the graph from Lemma 3.9, and denote $g = |V(J)|$. Apply Lemma 3.8 and let $\mathcal{A} = (C, \mathcal{E})$ be the resulting 3-uniform hypergraph, with $q = |C|$. Construct F_M as follows. Make C a clique. For every hyperedge $e \in \mathcal{E}$, add a fresh copy J_e of J. There are no edges between distinct copies J_e and J_f. For $x \in V(J_e)$ and $c \in C$, add the edge xc exactly when $c \notin e$. Thus every vertex of J_e is adjacent to all central vertices except the three vertices of e.

For $i \in [M]$, define

$$P_i = \bigcup_{e\in\mathcal{C}_i} V(J_e).$$

The ports are non-empty and pairwise disjoint because the families $\mathcal{C}_i$ are pairwise edge-disjoint. Also

$$|V(F_M)| = |C| + g|\mathcal{E}| = O(M^2)$$

and, since each $\mathcal{C}_i$ partitions C into triples,

$$|P_i| = g|\mathcal{C}_i| = O(M).$$

We next verify that no vertex of F_M is complete to any port. Fix $i \in [M]$. For $c \in C$, the partition $\mathcal{C}_i$ contains some triple e with $c \in e$, and every vertex of $J_e \subseteq P_i$ is non-adjacent to c. For $x \in V(J_f)$ for some $f \in \mathcal{E}$, since $q \geq 6$, $\mathcal{C}_i$ contains at least two triples. Choose $e \in \mathcal{C}_i$ with $e \neq f$. There are no edges between distinct gadget copies, so x has no neighbors in $J_e \subseteq P_i$.

It remains to prove the rigidity property. Let B be bipartite, and form X from the disjoint union of F_M and B by making every $b \in V(B)$ complete to the ports indexed by $A(b) \subseteq [M]$, with no other edges between B and F_M. Let $\varphi\colon F_M \to X$ be a homomorphism.

First, C is the unique q-clique of X, since inside F_M, any clique not contained in C uses vertices from at most one copy J_e. Because J is triangle-free, it uses at most two vertices from J_e, and it can also use only central vertices from $C \setminus e$. Hence its size is at most

$$2 + |C \setminus e| = q - 1.$$

If a clique uses vertices of B, then it uses no central vertex, because no vertex of B is adjacent to C. Moreover B has clique number at most 2, and the clique can use vertices from at most one copy J_e, again at most two of them. Thus such a clique has size at most $4 < q$. Therefore C is the unique q-clique of X.

Since the image of the clique C under φ is a q-clique in the loopless graph X, we have $\varphi(C) = C$. Let $\sigma = \varphi|_C$, viewed as a permutation of C.

Fix $e \in \mathcal{E}$. Every vertex of J_e is adjacent to every central vertex in $C \setminus e$, so every image of a vertex of J_e is adjacent to every vertex of

$$\varphi(C \setminus e) = C \setminus \sigma(e).$$

Now consider different possibilities for φ on J_e. No vertex of B qualifies, because vertices of B have no neighbors in C. A central vertex $c \in C$ qualifies exactly when $c \in \sigma(e)$, since the graph is loopless. A gadget vertex in J_f qualifies exactly when

$$C \setminus \sigma(e) \subseteq C \setminus f,$$

or equivalently $f \subseteq \sigma(e)$. Since both sets have size 3, this means $f = \sigma(e)$.

Consequently, φ can map a vertex of J_e to either the part of central clique K_3 on $\sigma(e)$ or to $J_{\sigma(e)}$ (if hyperedge $\sigma(e)$ exists, i.e. $\sigma(e) \in \mathcal{E}$). Note that there are no edges between the central triple $\sigma(e)$ and $J_{\sigma(e)}$.

The graph J_e is connected, so its image lies in one connected component of this common-neighbor set. It cannot lie in the central K_3, because there is no homomorphism $J \to K_3$. Hence $\sigma(e) \in \mathcal{E}$ and

$$\varphi(V(J_e)) \subseteq V(J_{\sigma(e)}).$$

Thus $\sigma(e) \in \mathcal{E}$ for every $e \in \mathcal{E}$. Since σ is a permutation of C, this gives $\sigma(\mathcal{E}) = \mathcal{E}$, so σ is an automorphism of the hypergraph $\mathcal{A}$, therefore $\sigma = \mathrm{id}_C$. It follows that φ fixes C pointwise and maps every copy J_e isomorphically onto itself, since the map $J_e \to J_{\sigma(e)}$ is an endomorphism and hence it is an automorphism.

Therefore φ is an automorphism of F_M whose image is contained in F_M. Finally, every port is a union of whole copies J_e,

$$P_i = \bigcup_{e \in \mathcal{C}_i} V(J_e),$$

and each such copy is mapped onto itself. Hence $\varphi(P_i) = P_i$ for every $i \in [M]$. $\square$

For a vertex z outside the frame and a set $A \subseteq [M]$, we say that z has anchor set A, meaning that z is adjacent to every vertex in every port P_i with $i \in A$, and to no other vertex of the frame.

Corollary 3.11. *Let X be as in Lemma 3.10. Let X' contain a copy F'_M of F_M, and let $\phi\colon X' \to X$ be a homomorphism. Then ϕ maps every port of F'_M onto the corresponding port of F_M. Hence, if a vertex $z \in V(X') \setminus V(F'_M)$ is complete to the ports indexed by A, then $\phi(z)$ is complete to the same ports in X.*

Proof. By Lemma 3.10, the restriction of ϕ to F'_M is a port-preserving isomorphism onto the target copy of F_M. Therefore every port is mapped onto its corresponding port. If z is complete to the ports indexed by A, then for every $i \in A$ and every $p \in P_i$, the edge zp of X' maps to an edge between $\phi(z)$ and $\phi(p)$. Since $\phi(P_i) = P_i$, the vertex $\phi(z)$ is complete to P_i for every $i \in A$. $\square$

Compiling the lists. We now convert the marked list-homomorphism instance into an ordinary homomorphism instance. The needed property for the reduction is that every list is defined by containment of a non-empty marker set over the same $M = O(\log r)$ markers.

Lemma 3.12. *There are graphs G_r and H_r such that*

$$(\Lambda_r, \Gamma_r, \mathcal{L}_r) \text{ has a list homomorphism} \iff G_r \to H_r.$$

Moreover, $|V(G_r)| = |V(\Lambda_r)| + O((\log r)^2)$ and $|V(H_r)| = |V(\Gamma_r)| + O((\log r)^2)$.

Proof. Take the frame F_M from Lemma 3.10, and identify its ports with the M markers. Construct H_r from the disjoint union of F_M and Γ_r by making every target vertex $x \in V(\Gamma_r)$ complete to precisely the ports in $A_\Gamma(x)$. Construct G_r from the disjoint union of a copy F'_M of the frame and Λ_r by making every source vertex $v \in V(\Lambda_r)$ complete to precisely the ports in $A_\Lambda(v)$. No other edges are added. The graph Γ_r is bipartite with sides S_E and $S_B \cup T$: all state and testing edges run between these two sides, and there are no edges inside S_B, inside S_E, inside T, or between S_B and T. Thus the target has the form required by Lemma 3.10 and Corollary 3.11.

If $\psi\colon \Lambda_r \to \Gamma_r$ is a list homomorphism, extend it by any fixed isomorphism $F'_M \to F_M$. The edges of Λ_r are preserved because ψ is a homomorphism. For every source vertex $v \in V(\Lambda_r)$, the list condition gives $A_\Lambda(v) \subseteq A_\Gamma(\psi(v))$, so all frame edges incident with v are also preserved. Thus the extension is a homomorphism $G_r \to H_r$.

Conversely, let $\phi\colon G_r \to H_r$ be a homomorphism. By Corollary 3.11, the copied frame maps portwise to the target frame. Every marker set $A_\Lambda(v)$ is non-empty, and no frame vertex is complete to any port, so $\phi(v)$ cannot be a frame vertex for $v \in V(\Lambda_r)$. Hence $\phi(v) \in V(\Gamma_r)$. Because v is complete to every port in $A_\Lambda(v)$ and the frame ports are preserved, $\phi(v)$ is complete to every corresponding target port. By construction of H_r, this means $A_\Lambda(v) \subseteq A_\Gamma(\phi(v))$, so $\phi(v) \in \mathcal{L}_r(v)$. Finally, every edge of Λ_r maps to an edge of Γ_r, since the only non-frame edges of H_r inside $V(\Gamma_r)$ are the edges of Γ_r. Thus $\phi|_{V(\Lambda_r)}$ is a list homomorphism.

The size bounds follow from $|V(F_M)| = O(M^2) = O((\log r)^2)$. $\square$

Corollary 3.13. *For every input graph Q of maximum degree at most 4,*

$$Q \text{ is 3-colorable} \iff G_r \to H_r.$$

Lemma 3.14.

$$|V(G_r)| = O(N/r + r^2).$$

Proof. This follows from Lemma 3.4 and $|V(F_M)| = O((\log r)^2) = O(r^2)$. □

Lemma 3.15.

$$|V(H_r)| \le 2^{O(r^2)}.$$

Proof. By Lemma 3.3 and Lemma 3.12,

$$|V(H_r)| = |V(\Gamma_r)| + O((\log r)^2) \le 2^{O(r^2)}.$$

□

3.5 Finishing the Lower Bound

Target degeneracy. Recall that degeneracy is the least d for which the vertices can be ordered so that each has at most d neighbors later in the order. We eliminate the edge-states first: although there may be $2^{O(r^2)}$ of them, each sees only the bucket-states of the labels it stores and the tests for its own edge indices, so it has degree $O(r)$ (Lemma 3.16). Once the edge-states are gone, every remaining non-frame vertex sees only its $O((\log r)^2)$ frame anchors (Lemma 3.17), and the frame itself has only $O((\log r)^2)$ vertices. The whole order therefore witnesses degeneracy $O(r)$. This upper bound is the only degeneracy estimate used in the lower-bound proof.

Lemma 3.16. *Every edge-state has degree $O(r)$ in H_r.*

Proof. Let $W = W_{Z,c,\Pi} \in S_E$. Its neighbors in S_B are exactly the bucket-states U_{ℓ,c_ℓ} with $\ell \in Z$. Hence

$$|N(W) \cap S_B| \le |Z| \le 2r.$$

For every edge index in $\mathrm{dom}(\Pi)$, the edge-state has at most four metadata neighbors and at most three color-check neighbors. Therefore

$$|N(W) \cap T| \le 7|\,\mathrm{dom}(\Pi)| \le 7r.$$

Its frame neighbors lie only in the port ρ_W. Since that port has size $O(\log r)$,

$$|N(W) \cap V(F_M)| = O(\log r).$$

Thus $\deg_{H_r}(W) = O(r)$. □

Lemma 3.17. *In $H_r - S_E$, every vertex of $S_B \cup T$ has degree $O((\log r)^2)$.*

Proof. After deleting S_E, there are no state edges and no testing edges left. Every bucket-state or test vertex is adjacent only to $O(\log r)$ frame ports, and every port has size $O(\log r)$. □

Theorem 3.18.

$$\mathrm{degen}(H_r) = O(r).$$

Proof. Use the following elimination order. First remove all vertices of S_E in an arbitrary order. By Lemma 3.16, each such vertex has at most $O(r)$ surviving neighbors when it is removed.

Then remove all vertices of $S_B \cup T$. By Lemma 3.17, after the edge-states are deleted their residual degree is $O((\log r)^2) = O(r)$.

Finally remove the frame. The frame has $O((\log r)^2) = O(r)$ vertices, so every remaining frame vertex has residual degree at most $O(r)$. Hence every vertex is removed with at most $O(r)$ later neighbors. ☐

ETH consequence.

Theorem 3.1 (Formal version of Theorem 1.1)**.** *Unless* ETH *fails, for every nondecreasing unbounded function $D\colon \mathbb{N} \to \mathbb{N}$ satisfying $D(n) = O(n^{1/3})$ and $D(2n) = O(D(n))$, there is no algorithm that solves* HOM *on every instance (G, H) satisfying* $\mathrm{degen}(H) \le D(n)$ *in time*

$$O^*\left(2^{o(D(n)n)}\right).$$

More precisely, for every sufficiently large integer r and all sufficiently large N, every N-vertex graph Q of maximum degree at most 4 can be transformed into an instance (G_r, H_r) of HOM*, in time polynomial in $|Q| + |H_r|$, with*

$$|V(G_r)| = O(N/r + r^2), \qquad |V(H_r)| = 2^{O(r^2)},$$

and with

$$\mathrm{degen}(H_r) = O(r),$$

such that Q is 3-colorable if and only if there is a homomorphism $G_r \to H_r$.

Proof of Theorem 3.1. Fix a nondecreasing unbounded function $D\colon \mathbb{N} \to \mathbb{N}$ satisfying $D(n) = O(n^{1/3})$ and $D(2n) = O(D(n))$. Suppose that an algorithm with the claimed running time exists on every instance (G, H) satisfying $\mathrm{degen}(H) \le D(n)$. Let Q be an N-vertex graph of maximum degree at most 4, and choose m minimally so that $mD(m) \ge N$. Monotonicity and $D(2m) = O(D(m))$ imply $mD(m) = \Theta(N)$. Let $\kappa \ge 1$ be a constant from Theorem 3.18 such that $\mathrm{degen}(H_r) \le \kappa r$, and set $r = \lfloor D(m)/\kappa \rfloor$. Since D is unbounded and $D(m) = O(m^{1/3})$, we have $r \to \infty$ and $r^3 = O(m) = o(N)$, so the construction yields

$$|V(G_r)| = O(N/D(m) + D(m)^2) = O(m), \qquad |V(H_r)| = 2^{O(D(m)^2)} = 2^{o(N)}.$$

Choose a constant integer c such that $|V(G_r)| \le cm$ for all sufficiently large N, and pad G_r with isolated vertices to a graph $\widehat{G}_r$ on $n = cm$ vertices. Since H_r has an edge, this padding preserves whether a homomorphism exists, while monotonicity and repeated use of the doubling condition give

$$\mathrm{degen}(H_r) \le \kappa r \le D(m) \le D(n), \qquad D(n)n = O(D(m)m) = O(N).$$

Thus the assumed algorithm, together with the construction, solves Q in time $2^{o(N)}$, contradicting Theorem 2.4. ☐

4 The Degeneracy Threshold from One to Two

This section proves the super-exponential lower bound for $\mathsf{HOM}(\cdot, \mathcal{H})$ even when $\mathcal{H}$ has constant degeneracy. The reduction uses the following bounded-alphabet 2-CSP lower bound of [Mar10, KMPS24].

Theorem 4.1 ([KMPS24, Theorem 1.6])**.** *Let f be a nondecreasing, unbounded, and polynomial-time computable function. Unless* ETH *fails, there is no algorithm running in time*

$$f(k-1)^{o(k/\log k)}$$

that decides all 2-CSP *instances whose constraint graph is a simple bipartite* 3*-regular graph on k vertices and whose alphabet size is smaller than $f(k)$.*

Theorem 4.2 (Formal version of Theorem 1.2)**.** *Let f be a nondecreasing, unbounded, and polynomial-time computable function. There is a polynomial-time reduction*[5] *that maps every* 2-CSP *instance covered by Theorem 4.1, whose constraint graph has k vertices, to an equivalent instance (G, H) of* HOM *with $N = |V(G)| = \Theta(k)$ and*

$$\operatorname{degen}(H) \leq 2, \qquad |V(H)| + |E(H)| = O(kf(k)^2),$$

for which no algorithm with running time

$$O^*\left(f(k-1)^{o(k/\log k)}\right)$$

exists, unless ETH *fails.*

At a high level, the proof first reduces a sparse CSP instance Γ from Theorem 4.1 to $\mathsf{HOM}(G_\Gamma, H_\Gamma)$. Then, we take a subdivision of both graphs (i.e., we replace every edge of both graphs by a path of length three). This fixed subdivision preserves the existence of a homomorphism and makes the target 2-degenerate.

However, the price of this step is an increase in size. After subdivision, the source has $|V(G_\Gamma)| + O(|E(G_\Gamma)|)$ vertices, and the target has $|V(H_\Gamma)| + O(|E(H_\Gamma)|)$ vertices. Thus the subdivision can be used in a fine-grained lower bound only when the source graph produced by the reduction is sparse.

4.1 Sparse Pinning Frame

We begin with the construction of a frame from [FGKM15] and then state its properties. For an integer $s \geq 1$, define a graph F_s as follows. It has $s+1$ distinguished vertices $z_0, z_1, \ldots, z_s$. For every $i \in [s]$, add four more vertices a_i, b_i, c_i, d_i. Add the edges of the 5-cycle

$$z_{i-1}a_i, \quad a_ib_i, \quad b_ic_i, \quad c_id_i, \quad d_iz_{i-1},$$

and add the five edges from z_i to this cycle,

$$z_iz_{i-1}, \quad z_ia_i, \quad z_ib_i, \quad z_ic_i, \quad z_id_i.$$

Let $Z_s = \{z_1, \ldots, z_s\}$ be the set of these cycle-completing vertices. Figure 4 gives an example.

[5] Here, polynomial time means polynomial in the combined size of the input and the output.

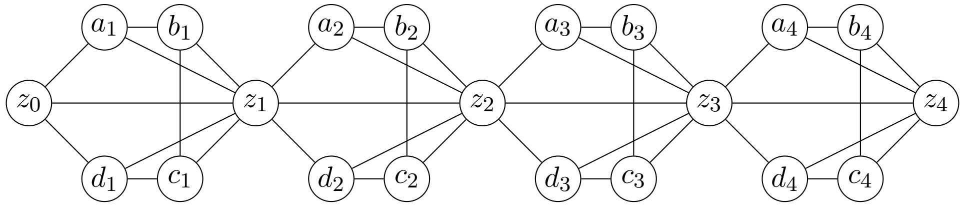

Figure 4: The graph F_4.

Lemma 4.3. *The graph F_s has degeneracy at most 3. Moreover, the following statements hold.*

1. *If $x \notin Z_s$, then $F_s[N_{F_s}(x)]$ is bipartite.*
2. *If $x = z_i \in Z_s$, then $F_s[N_{F_s}(z_i)]$ contains the 5-cycle $C_i = z_{i-1}a_ib_ic_id_iz_{i-1}$ and is non-bipartite.*
3. *For every $i \in [s]$, the unique non-bipartite connected component of $F_s[N_{F_s}(z_i)]$ is C_i. If $i \geq 2$, the only vertex of Z_s in this component is z_{i-1}; if $i = 1$, this component contains no vertex of Z_s.*

Proof. Every vertex among a_i, b_i, c_i, d_i has degree exactly 3 in F_s. Hence every subgraph containing one of these vertices has a vertex of degree at most 3. If a subgraph contains no such vertex, then it is a subgraph of the path on $z_0, z_1, \ldots, z_s$, and hence has a vertex of degree at most 1 unless it is empty. Therefore F_s is 3-degenerate.

The neighborhood of every vertex outside Z_s is bipartite by direct inspection. For example, $N(z_0) = \{a_1, d_1, z_1\}$ induces the path $a_1 - z_1 - d_1$, and $N(a_i) = \{z_{i-1}, b_i, z_i\}$ induces a path with center z_i. The cases b_i, c_i, d_i are the same up to symmetry along the cycle.

Inside $N(z_i)$, the vertices of C_i induce exactly that 5-cycle. The other frame neighbors of z_i, when they exist, lie in the three-vertex path $a_{i+1} - z_{i+1} - d_{i+1}$, which is disjoint from C_i inside the open neighborhood. Thus C_i is the unique non-bipartite connected component. For $i \geq 2$, the only vertex of Z_s among the vertices of C_i is z_{i-1}; for $i = 1$, the cycle C_i contains no vertex of Z_s. □

The next step is to use the distinguished vertices in Z_s as pin locations for the types that occur in the CSP reduction. In that reduction, CSP variables and CSP edges both serve as types: for a CSP variable v and a possible value a, the vertex $A_{v,a}$ has type v. For a CSP edge $e = (u, v)$ and an allowed pair (a, b), where a corresponds to u and b corresponds to v, the vertex $R_{e,a,b}$ has type e. The vertex $R_{e,a,b}$ is adjacent to the pin p_e and to the value vertices $A_{u,a}$ and $A_{v,b}$, whose types are u and v, respectively; both are different from the constraint type e.

Let T be a finite set of types, let $L = |T|$, put $s = 10L + 10$ and construct F_s, then enumerate $T = \{t_1, \ldots, t_L\}$. For t_j, define two pins

$$p_{t_j} = z_{10j}, \qquad q_{t_j} = z_{10j+5}.$$

The spacing ensures that all pins are distinct and that the two pins of any type have no common frame neighbor. For an edge type e, only the pin p_e will be used by constraint vertices.

We say that a graph is obtained from F_s by safe attachments if new vertices are added only in the following two forms. A value vertex of type t is a new vertex adjacent to the two pins p_t and q_t. A constraint vertex of type t is a new vertex adjacent to the pin p_t and to two value vertices whose types are different from t. See Figure 5.

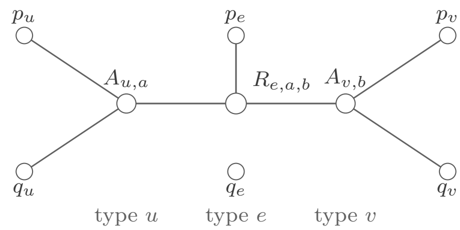

Figure 5: A local view of safe attachments.

Lemma 4.4. *Let H be any graph obtained from F_s by safe attachments. Then every homomorphism $\psi \colon F_s \to H$ fixes every vertex of Z_s:*

$$\psi(z_i) = z_i, \qquad \textit{for all } i \in [s].$$

Proof. First, the only vertices of H whose open neighborhoods are non-bipartite are $z_1, \dots, z_s$. Indeed, frame vertices outside Z_s keep the same open neighborhoods as in F_s, because attachments are made only to vertices of Z_s. A value vertex has a bipartite open neighborhood: apart from constraint vertices, it sees only its two nonadjacent pins, and the constraint vertices adjacent to it are not adjacent to those pins. A constraint vertex has an independent open neighborhood consisting of one pin and two value vertices.

Moreover, for $z_i \in Z_s$, the added vertices in $N_H(z_i)$ are isolated in the induced graph $H[N_H(z_i)]$. If an added value vertex is adjacent to z_i, then its other pin is not adjacent to z_i by the spacing of pins. If an added constraint vertex is adjacent to z_i, then its other neighbors are two value vertices, none of which is adjacent to z_i. Consequently, the unique non-bipartite component of $H[N_H(z_i)]$ is still the cycle component C_i from Lemma 4.3.

Let $\psi \colon F_s \to H$ be a homomorphism. For each $i \in [s]$, the cycle C_i is contained in $N_{F_s}(z_i)$. Its image is an odd closed walk in $H[N_H(\psi(z_i))]$. Hence $H[N_H(\psi(z_i))]$ is non-bipartite, and therefore $\psi(z_i) \in Z_s$. Let $\psi(z_i) = z_{r_i}$ with $r_i \in [s]$.

For $i \geq 2$, the cycle C_i contains the vertex $z_{i-1} \in Z_s$. Since C_i is connected and contains an odd cycle, its image lies in the unique non-bipartite component C_{r_i} of $H[N_H(z_{r_i})]$. The vertex z_{i-1} maps to a vertex of Z_s, and the only vertex of Z_s in C_{r_i} is $z_{r_i - 1}$. Thus $r_i \geq 2$ and $\psi(z_{i-1}) = z_{r_i-1}$. Equivalently, $r_{i-1} = r_i - 1$ for every $i = 2, \dots, s$. Since $1 \leq r_s \leq s$, we get $r_1 \leq 1$. Also $r_1 \geq 1$, so $r_1 = 1$ and $r_i = i$ for every $i \in [s]$. ☐

4.2 Reduction

Let

$$\Gamma = (P, \Sigma, \{C_e\}_{e \in E(P)})$$

be a 2-CSP instance whose constraint graph P is a simple bipartite 3-regular graph on k vertices. Let $q = |\Sigma|$. Since P is 3-regular, $|E(P)| = 3k/2$. Fix an arbitrary orientation of every edge $e \in E(P)$.

Define the type set

$$T = V(P) \cup E(P).$$

Then $|T| = k + 3k/2 = 5k/2$. Build the pinning frame F_s with $s = 10|T| + 10$ and with the pin pairs (p_t, q_t) for $t \in T$ as above.

The source graph G_Γ starts with a copy of F_s. For every CSP variable $v \in V(P)$, add a vertex x_v and the two pin edges $x_v p_v$ and $x_v q_v$. For every oriented constraint edge $e = (u, v) \in E(P)$, add a vertex y_e and the three edges

$$y_e p_e, \qquad y_e x_u, \qquad y_e x_v.$$

The target graph H_Γ starts with a copy of the same frame F_s. For every CSP variable $v \in V(P)$ and every value $a \in \Sigma$, add a value vertex $A_{v,a}$ and the two pin edges $A_{v,a}p_v$ and $A_{v,a}q_v$. For every oriented constraint edge $e = (u, v)$ and every allowed pair $(a, b) \in C_e$, add a constraint vertex $R_{e,a,b}$ and the three edges

$$R_{e,a,b}p_e, \qquad R_{e,a,b}A_{u,a}, \qquad R_{e,a,b}A_{v,b}.$$

Lemma 4.5. *The* CSP *instance* Γ *is satisfiable if and only if* $G_\Gamma \to H_\Gamma$*.*

Proof. Suppose first that Γ has a satisfying assignment $\alpha \colon V(P) \to \Sigma$. Map the frame in G_Γ identically to the frame in H_Γ. For every variable vertex x_v, set $\phi(x_v) = A_{v,\alpha(v)}$. For every constraint edge $e = (u, v)$, the pair $(\alpha(u), \alpha(v))$ belongs to C_e, so the constraint vertex $R_{e,\alpha(u),\alpha(v)}$ exists; set $\phi(y_e) = R_{e,\alpha(u),\alpha(v)}$. All frame edges, pin edges, and incidence edges are preserved by construction, so ϕ is a homomorphism.

Conversely, let $\phi \colon G_\Gamma \to H_\Gamma$ be a homomorphism. The restriction of ϕ to the frame copy in G_Γ fixes every vertex of Z_s by Lemma 4.4. In particular, all pins are fixed.

For a CSP variable v, the vertex x_v is adjacent to both p_v and q_v. Therefore $\phi(x_v)$ is a common neighbor of p_v and q_v in H_Γ. The pin pair has no common neighbor inside the frame, and the only added common neighbors are the value vertices $A_{v,a}$ with $a \in \Sigma$. Thus there is a unique value $\alpha(v) \in \Sigma$ such that $\phi(x_v) = A_{v,\alpha(v)}$.

Now let $e = (u, v)$ be a constraint edge. Since y_e is adjacent to p_e, x_u, and x_v, the vertex $\phi(y_e)$ is a common neighbor of p_e, $A_{u,\alpha(u)}$, and $A_{v,\alpha(v)}$. No frame vertex has this property: a frame neighbor of $A_{u,\alpha(u)}$ must be p_u or q_u, while a frame neighbor of $A_{v,\alpha(v)}$ must be p_v or q_v, and these four pins are distinct by construction because $u \neq v$. No value vertex has this property either, since value vertices have no value neighbors. Hence $\phi(y_e)$ is a constraint vertex. A constraint vertex adjacent to p_e has type e, and its adjacency to $A_{u,\alpha(u)}$ and $A_{v,\alpha(v)}$ forces this vertex to be $R_{e,\alpha(u),\alpha(v)}$. Therefore $(\alpha(u), \alpha(v)) \in C_e$. This holds for every constraint edge, so α satisfies Γ. ☐

Lemma 4.6. *The reduction takes* $\Gamma = (P, \Sigma, \{C_e\}_{e \in E(P)})$*, constructs* G_Γ *and* H_Γ *in time polynomial in the output size and satisfies*

$$|V(G_\Gamma)| = \Theta(k), \qquad |E(G_\Gamma)| = O(k)$$

and

$$|V(H_\Gamma)| + |E(H_\Gamma)| = O(kq^2),$$

where $k = |V(P)|$*, and* $q = |\Sigma|$*.*

Proof. The frame has $|V(F_s)| = (s + 1) + 4s = 5s + 1$ vertices, where $s = 10|T| + 10 = 25k + 10$, so $|V(F_s)| = \Theta(k)$. The source graph adds one vertex for every vertex of P and one vertex for every edge of P, hence adds $k + 3k/2 = O(k)$ vertices. Therefore $|V(G_\Gamma)| = \Theta(k)$. The source graph has the $O(k)$ frame edges, two pin edges for every CSP variable, and three edges for every CSP edge vertex. Thus $|E(G_\Gamma)| = O(k)$.

The target graph has the frame, kq value vertices, and at most

$$\sum_{e\in E(P)} |C_e| \le |E(P)|q^2 = O(kq^2)$$

constraint vertices. Each added value vertex contributes two pin edges, and each constraint vertex contributes three edges. The frame has $O(k)$ edges. Therefore $|V(H_\Gamma)| + |E(H_\Gamma)| = O(kq^2)$, and the construction time is polynomial in this output size. $\square$

For a graph X, let $S_3(X)$ denote the graph obtained from X by replacing every edge by a path of length 3.

Lemma 4.7. *For all finite loopless graphs G and H,*

$$G \to H \qquad \iff \qquad S_3(G) \to S_3(H).$$

Proof. The forward implication is immediate. A homomorphism $\varphi\colon G \to H$ maps each edge $uv \in E(G)$ to an edge $\varphi(u)\varphi(v) \in E(H)$, and the subdivided path replacing uv can be mapped to the subdivided path replacing $\varphi(u)\varphi(v)$.

For the reverse implication, choose an arbitrary orientation of each edge of H. If an oriented edge ab is replaced in $S_3(H)$ by the path $a - x_{ab} - y_{ab} - b$, define a map $\rho\colon V(S_3(H)) \to V(H)$ by

$$\rho(a) = a \text{ for original vertices } a, \qquad \rho(x_{ab}) = b, \qquad \rho(y_{ab}) = a.$$

This map sends every edge of $S_3(H)$ to an edge of H.

We claim that if $w_0w_1w_2w_3$ is a walk of length 3 in $S_3(H)$, then $\rho(w_0)\rho(w_3) \in E(H)$. Indeed, every two-step walk centered at an internal subdivision vertex has endpoints with the same ρ-image. If w_1 is internal, then $\rho(w_0) = \rho(w_2)$, and the edge w_2w_3 gives $\rho(w_2)\rho(w_3) \in E(H)$. If w_1 is original, then w_2 is internal, so $\rho(w_1) = \rho(w_3)$, and the edge w_0w_1 gives $\rho(w_0)\rho(w_1) \in E(H)$. This proves the claim.

Now let $\psi\colon S_3(G) \to S_3(H)$ be a homomorphism. For every original vertex $v \in V(G)$, define $\varphi(v) = \rho(\psi(v))$. If $uv \in E(G)$, then the subdivided path replacing uv in $S_3(G)$ maps under ψ to a walk of length 3 in $S_3(H)$ from $\psi(u)$ to $\psi(v)$. Hence, $\varphi(u)\varphi(v) \in E(H)$ and $\varphi\colon G \to H$ is a homomorphism. $\square$

Lemma 4.8. *For every graph H, the graph $S_3(H)$ satisfies*

$$\operatorname{degen}(S_3(H)) \le 2.$$

Proof. Let J be a non-empty subgraph of $S_3(H)$. If J contains an internal subdivision vertex, then that vertex has degree at most 2 in $S_3(H)$, and hence degree at most 2 in J. If J contains no internal subdivision vertex, then J is an independent set of original vertices. Thus every non-empty subgraph of $S_3(H)$ has a vertex of degree at most 2. $\square$

Proof of Theorem 4.2. Apply Theorem 4.1 with the function f from the statement, and let Γ be an instance with constraint graph on k vertices and alphabet size $q < f(k)$.

Construct (G_Γ, H_Γ) as in Section 4.2, in time polynomial in the output size by Lemma 4.6, and then set

$$\widehat{G}_\Gamma = S_3(G_\Gamma), \qquad \widehat{H}_\Gamma = S_3(H_\Gamma).$$

By Lemma 4.5 and Lemma 4.7, Γ is satisfiable if and only if $\widehat{G}_\Gamma \to \widehat{H}_\Gamma$. By Lemma 4.8, the target satisfies $\mathrm{degen}(\widehat{H}_\Gamma) \le 2$. Let $N = |V(\widehat{G}_\Gamma)|$. By Lemma 4.6,

$$N = |V(G_\Gamma)| + 2|E(G_\Gamma)| = \Theta(k).$$

The same size bound gives

$$|V(\widehat{H}_\Gamma)| + |E(\widehat{H}_\Gamma)| = O(|V(H_\Gamma)| + |E(H_\Gamma)|) = O(kq^2) = O(kf(k)^2).$$

The construction time is polynomial in this output size. Therefore, an algorithm solving all constructed instances in time

$$f(k-1)^{o(k/\log k)} \cdot (|V(G)| + |V(H)|)^{O(1)}$$

would give an $f(k-1)^{o(k/\log k)}$-time algorithm for the CSP instances covered by Theorem 4.1, contradicting that theorem. □

5 No-Compression Barriers for SAT Reductions

Theorem 1.1 shows that degeneracy is visible linearly in the exponent of HOM, but the hard targets behind it are large: the construction needs a target of size $h = 2^{\Theta(D^2)}$ to realize degeneracy scale D. This is polynomial in n only while $D = O(\sqrt{\log n})$; for larger degeneracy the target becomes superpolynomial. It is therefore natural to ask whether the same linear dependence can be forced while the target stays polynomially bounded.

This section gives a conditional reason to expect that it cannot, at least not through the standard route. Fine-grained lower bounds for HOM, like those for many neighboring problems, are ultimately reductions from sparse 3-SAT via ETH and the Sparsification Lemma. We prove that a polynomial target reduction would "compress" the SAT instance, thereby contradicting Hypothesis 1.3.

The argument has four parts. First, we discuss why an explicit sparse 3-SAT instance carries hardness only at scale $s/\log s$, where s is its bit length, and we recall the No-Compression Hypothesis stating no ordinary reduction does better. Second, we apply this argument to sparse CSP instances with a linear number of constraints, showing that one cannot improve the known lower bounds for it under Hypothesis 1.3. Third, we show that degenerate targets compress the boundary information of homomorphism into short list records; this lets us reduce HOM to another problem which is compressible. Finally, we show the resulting no-compression barrier for HOM.

5.1 Sparse-SAT Information Scale

The Sparsification Lemma [IPZ01] lets one run ETH through sparse 3-SAT: formulas on N variables with $O(N)$ clauses still require $2^{\Omega(N)}$ time under ETH. If the problem is measured by an explicit bit length s, then to encode a sparse formula one needs $s = \Theta(N \log N)$. Hence the hardness of sparse 3-SAT is

$$2^{\Omega(N)} = 2^{\Omega(s/\log s)}.$$

under ETH.

To prove a lower bound *above* this scale for an s-bit target problem, a sparse-SAT reduction must do some compression of the formula. A broad variety of fine-grained lower bounds is known, yet to our knowledge not one of them establishes a bound better than $2^{\Omega(s/\log s)}$ in the input bit length s.

This distinction has a precedent in the barrier of Kulikov and Mihajlin for the Edge Coloring problem [KM24]. The input to this problem has size n^2 and the best known upper bound is of the form $2^{O(n^2)}$. Proving an α^{n^2} lower bound for this problem under SETH for any $\alpha > 0$ would break SETH. The hypothesis we use is the ETH analogue of that barrier, and it does not follow from their argument.

We now state the formal version of the no-compression principle used in the rest of the section.

Hypothesis 1.3 (No-Compression Hypothesis)**.** *Fix an explicit encoding of a decision problem $\mathcal{A}$, and let s be the bit length of an encoded instance. There is no fine-grained reduction from sparse* 3-SAT *on N variables to $\mathcal{A}$, and no function $\Psi(s) = \omega(s/\log s)$, with the following property: if $\mathcal{A}$ has an algorithm running in time $2^{o(\Psi(s))}$ on s-bit instances, then sparse* 3-SAT *has an algorithm running in time $2^{o(N)}$.*

Equivalently, there is no reduction establishing under ETH *a lower bound*

$$2^{\Omega(\Psi(s))}$$

for any $\Psi(s) = \omega(s/\log s)$.

5.2 Application to Sparse CSP

Theorem 5.1. *Assume Hypothesis 1.3. Let $k = k(n)$ and $h = h(n)$. Set $B(n) = k \log n + h^k$. Suppose that*

$$\log h = \omega\left(\frac{B(n)}{\log n + \log B(n)}\right).$$

Then no fine-grained reduction from sparse 3-SAT *can establish, under* ETH*, an $h^{\Omega(n)}$-time lower bound for sparse (k, h)-*CSP*.*

Proof. Consider a sparse (k, h)-CSP instance with n variables. Each constraint has arity at most k, and there are at most Cn constraints for some $C > 0$. Thus the constraint scopes can be encoded using $O_C(nk \log n)$ bits. Each constraint relation has arity at most k and is written as a truth table of size at most h^k. Since the number of constraints is at most Cn, all truth tables use $O_C(nh^k)$ bits. Hence the total bit length is

$$s = O_C(nB(n)).$$

Now suppose that a fine-grained reduction established an $h^{\Omega(n)}$ lower bound for this CSP family under ETH. By the assumption, $n \log h = \omega(nB(n)/\log(nB(n)))$. Since $s = O_C(nB(n))$, this implies $n \log h = \omega(s/\log s)$, contradicting Hypothesis 1.3. □

Note that for fixed k, the condition holds for every $h = h(n) \to \infty$ with $h^k = O(\log n)$. However, for constant h, one has $n \log h = \Theta(n)$ and $s/\log s = \Theta(n)$, so it does not rule out ETH lower bounds.

5.3 Succinct List-Extension Homomorphism Problem

Suppose a partial homomorphism has already mapped some neighbors of an unmapped source vertex. The vertices of H still available to that vertex are exactly the common neighborhood in H of the images of its mapped neighbors. The lemma says that in a degenerate target this common neighborhood is either small or is cut out by few constraints.

Lemma 5.2. *Let $D \in \mathbb{Z}_{\geq 0}$, and let H be a loopless graph with $\mathrm{degen}(H) \leq D$. For every set $C \subseteq V(H)$, define*

$$A(C) = \bigcap_{c \in C} N_H(c),$$

with the convention $A(\varnothing) = V(H)$. Then either

$$|A(C)| \leq D \qquad \text{or} \qquad |C| \leq D.$$

Proof. Suppose not. Then $|A(C)| \geq D + 1$ and $|C| \geq D + 1$. Since H is loopless, $A(C) \cap C = \varnothing$: if $c \in C$, then $c \notin N_H(c)$, so $c \notin A(C)$.

Choose subsets $A' \subseteq A(C)$ and $C' \subseteq C$ with $|A'| = |C'| = D + 1$. By definition of $A(C)$, every vertex of A' is adjacent to every vertex of C'. Thus H contains $K_{D+1,D+1}$ as a subgraph. This graph has minimum degree $D + 1$, and hence degeneracy would be at least $D + 1$. ☐

Succinct list extension is the bookkeeping problem for exactly the information that survives this compression. Each source vertex carries either the short list of colors still available to it or a short certificate of that list, and the task is to complete the partial homomorphism subject to these lists.

Definition 5.3. *Fix functions $D = D(n)$ and $T = T(n)$. An instance of $\mathsf{SLEH}_{D,T}$ consists of:*

1. *a source graph G on at most n vertices and at most Tn edges;*
2. *a target graph H with $h \leq n$ and $\mathrm{degen}(H) \leq D$;*
3. *for every $x \in V(G)$, a succinct list record of one of the following two forms:*
   (a) *an explicit list $S_x \subseteq V(H)$ with $|S_x| \leq D$, interpreted as $L(x) = S_x$;*
   (b) *an intersection certificate $C_x \subseteq V(H)$ with $|C_x| \leq D$, interpreted as*

$$L(x) = \bigcap_{c \in C_x} N_H(c).$$

The task is to decide whether there exists a homomorphism $\psi \colon G \to H$ such that $\psi(x) \in L(x)$ for every $x \in V(G)$.

Since $h \leq n$, a target vertex can be named using $O(\log n)$ bits, so each list record costs $O(D \log n)$ bits. The source edge set costs $O(Tn \log n)$ bits, and the target, being D-degenerate, has at most Dh edges and so is stored in $O(Dh \log h) = O(Dn \log n)$ bits. Hence every such instance has bit size

$$s = O((T + D)n \log n).$$

5.4 Reduction

Theorem 5.4. *Let $D = D(n) \geq 1$ and $\tau = \tau(n) \geq 1$. There is a fine-grained Turing reduction from* HOM *instances (G, H) with source size n and target size h satisfying*

$$h \leq n, \qquad \mathrm{degen}(H) \leq D$$

to $\mathsf{SLEH}_{D,\tau}$ *such that every oracle query has bit size*

$$s = O(n(D + \tau) \log n),$$

and the total number of oracle queries, as well as the total non-oracle work, is at most

$$2^{O\left(n \log D + \frac{n}{\tau}(\log h + \log \tau)\right)}.$$

Proof. The idea is a branching scheme that keeps the residual source graph sparse enough to hand to the oracle. We process the target in a degeneracy order. At each step we either certify that the unmapped part of G has no dense core, in which case it is sparse and we ship it to the list-extension oracle, or we locate a dense core and branch on its vertices.

Fix an ordering $x_1, \ldots, x_n$ of $V(G)$, and compute a degeneracy ordering $w_1, \ldots, w_h$ of H such that every vertex has at most D neighbors to its right. For each target index $i \in [h]$ let

$$R_i \coloneqq N_H(w_i) \cap \{w_{i+1}, \ldots, w_h\}, \qquad |R_i| \leq D.$$

Set $p_{\max} = \lfloor \frac{n}{\tau+1} \rfloor$. The reduction first iterates over all sets $P \subseteq V(G)$ with $|P| \leq p_{\max}$ and maps $\sigma \colon P \to [h]$. Now, fix some P and σ.

We then branch as follows. A state of the branch is a pair (α, i), where $M = \mathrm{dom}(\alpha)$ is the set of currently mapped source vertices, $\alpha \colon M \to V(H)$ and $i \in [h+1]$. At every state we reject if α is not a partial homomorphism: if there is an edge $uv \in E(G)$ with $u, v \in M$ and $\alpha(u)\alpha(v) \notin E(H)$, the branch is discarded.

Let

$$U = V(G) \setminus M$$

be the current unmapped set, and let

$$K = \mathrm{core}_\tau(G[U])$$

be the τ-core of the residual graph $G[U]$, i.e., the maximal induced subgraph of $G[U]$ of minimum degree at least τ.

If $K = \varnothing$, then $G[U]$ is $(\tau - 1)$-degenerate. Therefore $|E(G[U])| \leq \tau n$. We now output a succinct list-extension oracle query. Its source graph is $G[U]$. For every $y \in U$, define the set of boundary vertices

$$C_\alpha(y) = \{\alpha(x) \colon x \in M \cap N_G(y)\}.$$

The vertices available to y because of its already mapped neighbors are exactly

$$A_\alpha(y) = \bigcap_{c \in C_\alpha(y)} N_H(c).$$

If $|A_\alpha(y)| \le D$, give y the explicit list record $A_\alpha(y)$. If $|A_\alpha(y)| > D$, then Lemma 5.2 implies $|C_\alpha(y)| \le D$, and we give y the intersection-certificate record $C_\alpha(y)$. This query is equivalent to asking whether α extends to a full homomorphism $G \to H$.

Suppose now that $K \neq \varnothing$. If $i = h+1$, reject the branch. If there is no vertex $v \in P \cap V(K)$ with $\sigma(v) = i$, advance to state $(\alpha, i+1)$ and finish this branch.

If such vertices exist, choose the first one and call it v. Since $v \in K$, choose the first τ neighbors of v in K and call this set $S(v)$. Branch over all maps

$$\beta\colon S(v) \to R_i.$$

For each branch, set

$$\alpha' = \alpha \cup \{v \mapsto w_i\} \cup \beta.$$

If α' is inconsistent, discard it; otherwise recurse from (α', i). The index remains i because other core vertices may also map to w_i.

Soundness of the reduction is immediate from the construction of the list-extension query: if an oracle query accepts, the list homomorphism on $G[U]$ respects all residual edges and all boundary constraints to M, and therefore combines with α to give a homomorphism $G \to H$.

For completeness, suppose $\varphi\colon G \to H$ is a homomorphism. We construct P, σ and a branch that finds some homomorphism $G \to H$. We follow a branch maintaining the invariant:

$$\alpha(x) = \varphi(x) \text{ for every } x \in \operatorname{dom}(\alpha),$$

and every vertex of the current τ-core K is mapped by φ into the active suffix

$$\{w_i, w_{i+1}, \ldots, w_h\}.$$

The invariant holds at the root $M = \varnothing, \alpha = \varnothing$.

If $K = \varnothing$, the final oracle query is satisfied by $\varphi|_U$. If $K \neq \varnothing$ and no vertex of K maps to w_i, increase i. Otherwise choose the first vertex $v \in V(K)$ with $\varphi(v) = w_i$, add v to P, and set $\sigma(v) = i$. Let $S(v)$ be the first τ neighbors of v in K. For every $u \in S(v)$, the edge uv gives $\varphi(u) \in N_H(w_i)$; by the suffix invariant and looplessness, in fact $\varphi(u) \in R_i$. Thus the branch assigning $u \mapsto \varphi(u)$ for every $u \in S(v)$ is present. Extend α according to φ on $\{v\} \cup S(v)$ and continue with the same index i. Each pivot step maps $\tau + 1$ fresh vertices, so P has at most $p_{\max}$ vertices at the end. Hence some branch reaches an accepting oracle query for some (P, σ).

It remains to bound the oracle input bit size and the running time. At a leaf, $G[U]$ has fewer than τn edges, whose encoding costs $O(\tau n \log n)$ bits. The list records cost $O(Dn \log n)$ bits, and the D-degenerate target costs $O(Dh \log h) = O(Dn \log n)$ bits. Thus every query has bit size $O(n(D+\tau)\log n)$.

The number of pairs (P, σ) is at most

$$\sum_{p \le p_{\max}} \binom{n}{p} h^p \le 2^{O\left(n + \frac{n}{\tau}(\log h + \log \tau)\right)}.$$

For a fixed pair (P, σ), the branches contribute at most $D^n = 2^{O(n \log D)}$, because every source vertex is assigned at most once. This proves the stated bound. $\square$

5.5 No-Compression Barrier

Theorem 5.5 (Formal version of Theorem 1.4)**.** *Assume Hypothesis 1.3. Let* $h = h(n) \le n$ *and* $1 \le D(n) \le n$*, and let* $\Lambda(n) = n \cdot \lambda(n)$ *for some* $\lambda(n) = \omega(\max\{D(n), \sqrt{\log h}\})$*. Then there is no fine-grained reduction from sparse* 3-SAT *on* N *variables to* HOM *restricted to instances* (G, H) *with source size* n*,* $|V(H)| \le h$*, and* $\operatorname{degen}(H) \le D(n)$*, with the following property: if these instances admit an algorithm running in time* $2^{o(\Lambda(n))}$*, then sparse* 3-SAT *admits an algorithm running in time* $2^{o(N)}$*.*

Proof. Suppose toward contradiction that such a fine-grained reduction establishes the lower bound $\Lambda(n) = n \cdot \omega(\max\{D(n), \sqrt{\log h}\})$. Set $\tau = \lceil \max\{D(n), \sqrt{\log h}\} \rceil$, and apply Theorem 5.4. Hence every oracle query has bit size $s := O(n\tau \log n)$.

The overhead exponent is

$$O\left(n \log D + \frac{n}{\tau}(\log h + \log \tau)\right) = O(n \cdot \tau).$$

Since $\tau \le n$, each query size satisfies $s = O(n\tau \log n)$. Thus $\Lambda(n) = n\lambda(n) = \omega(n\tau) = \omega(s/\log s)$, while the reduction overhead is $2^{o(\Lambda(n))}$. Therefore an algorithm for the queried succinct list-extension instances with running time $2^{o(\Lambda(n))}$ would solve the promised HOM instances in time $2^{o(\Lambda(n))}$. Composing this with the assumed 3-SAT reduction would give a $2^{o(N)}$ algorithm for 3-SAT on N variables, contradicting ETH. Equivalently, the composed proof would establish a lower bound above the $s/\log s$ scale, contradicting Hypothesis 1.3. ☐

6 Generated Formulas and Hard Witnesses

This section shows that a strong refutation of Hypothesis 1.3 implies circuit lower bounds. The result of Impagliazzo, Paturi, and Zane [IPZ01] shows that 3-SAT remains hard even when the number of clauses is linear. Therefore, in some sense, it provides a generator that takes $O(n \log n)$ bits as input and outputs a 3-SAT formula on n variables that is hard under ETH. Probably, the most natural way to refute Hypothesis 1.3 is to construct a more efficient generator of hard 3-SAT formulas.

We show that if a short seed can generate hard formulas, then the generated formulas must sometimes have satisfying assignments of high circuit complexity. Those hard assignments can then be exposed one bit at a time by a language in P^{NP}. Any small circuit for this language, after hardwiring the generator seed, would give a small circuit for the hard satisfying assignment, which is impossible.

The proof has three steps. First we formally state what we mean by generators of hard formulas. Second we show that this generator forces satisfying assignments of high circuit complexity: otherwise one could enumerate all small circuits, view each as an assignment, and solve the generated formulas in $2^{o(N)}$ time. Finally we package the lexicographically first satisfying assignment into a language in P^{NP}, which is an easy part.

Definition 6.1. *Fix* $0 < \alpha \le 1$ *and set* $r_\alpha(N) = \lceil N^\alpha \rceil$*. An* α-compressed hard formula generator *is a uniform family of maps*

$$G_N : \{0,1\}^{r_\alpha(N)} \to \{3\text{-}CNF \text{ formulas over } N \text{ variables}\}$$

such that the following hold.

- $G_N(y)$ *is computable in time* $\mathrm{poly}(N)$*.*

- *If there is an algorithm* A *and a function* $t(N) = 2^{o(N)}$ *such that, for every sufficiently large* N *and every formula* $F \in \mathrm{Im}(G_N)$*, the algorithm* A *on input* F *decides whether it is satisfiable in time* $t(N)$*, then* ETH *is false.*

Equivalently, assuming ETH*,* SAT *restricted to* $\mathrm{Im}(G_N)$ *has no uniform* $2^{o(N)}$*-time algorithm.*

Lemma 6.2. *Assume* ETH*, and let* $G = \{G_N\}$ *be an* α*-compressed hard formula generator. Let* $s\colon \mathbb{N} \to \mathbb{N}$ *be a polynomial-time computable function satisfying*

$$s(N)\log(s(N) + \log N) = o(N).$$

Then, for infinitely many N*, there is a seed* $y_N \in \{0,1\}^{r_\alpha(N)}$ *such that* $G_N(y_N)$ *is satisfiable, but every satisfying assignment* $a \in \{0,1\}^N$ *of* $G_N(y_N)$ *satisfies* $\mathsf{CC}[a] > s(N)$*.*

Proof. Suppose the conclusion fails. Then, for all sufficiently large N, every satisfiable formula in $\mathrm{Im}(G_N)$ has at least one satisfying assignment of circuit complexity at most $s(N)$.

This gives a $2^{o(N)}$-time algorithm for satisfiability on $\mathrm{Im}(G_N)$. On input $F \in \mathrm{Im}(G_N)$, enumerate all Boolean circuits on $\lceil \log_2 N \rceil$ inputs and of size at most $s(N)$. The number of such circuits is at most

$$2^{O(s(N)\log(s(N)+\log N))} = 2^{o(N)}.$$

For each circuit C, form the assignment $a_C = (C(0), C(1), \ldots, C(N-1))$ and test whether $F(a_C) = 1$. Since F has size $\mathrm{poly}(N)$, each test takes $\mathrm{poly}(N)$ time.

If some assignment a_C satisfies F, accept; otherwise reject. The algorithm is correct because a satisfiable generated formula has a small-circuit satisfying assignment by the contrary assumption, while an unsatisfiable formula has no satisfying assignment at all. Its running time is

$$2^{o(N)}\,\mathrm{poly}(N).$$

□

Fix an α-compressed hard formula generator $G = \{G_N\}$. We now turn satisfying assignments into a language by asking for one indexed bit of the lexicographically first witness. Use any fixed effective self-delimiting encoding of triples $\langle N, y, i\rangle$ such that, for fixed N and y, all choices of the index $i \in \{0,1\}^{\lceil \log_2 N\rceil}$ have the same total input length.

Definition 6.3. *The language* LexWit_G *is defined as follows. A valid input is a tuple* $\langle N, y, i\rangle$ *with* $y \in \{0,1\}^{r_\alpha(N)}$ *and* $i \in \{0,1\}^{\lceil \log_2 N\rceil}$*. Let* $F = G_N(y)$*.*

- *If* F *is unsatisfiable, then* $\mathsf{LexWit}_G(N, y, i) = 0$*.*

- *If* F *is satisfiable, let* $a^{N,y} \in \{0,1\}^N$ *be the lexicographically first satisfying assignment of* F*, then*

$$\mathsf{LexWit}_G(N, y, i) = \begin{cases} a_i^{N,y} & \text{if } i < N, \\ 0 & \text{if } i \geq N. \end{cases}$$

Let $\ell(N)$ denote the input bit size of LexWit_G, so $\ell(N) \coloneqq \Theta(N^\alpha)$.

Lemma 6.4. *For every fixed* $0 < \alpha \le 1$,

$$\mathsf{LexWit}_G \in \mathsf{P}^{\mathsf{NP}}.$$

Proof. On input $\langle N, y, i\rangle$, first check the encoding validity and reject invalid inputs. Let $F = G_N(y)$. Use an NP oracle to compute the lexicographically first satisfying assignment of F, if one exists. First ask whether F is satisfiable; if not, output 0. Otherwise, build the assignment bit by bit. For a prefix $p \in \{0,1\}^j$, the oracle query asks whether

$$\exists a \in \{0,1\}^N \quad a \text{ extends } p \text{ and } F(a) = 1.$$

Using at most $N+1$ such oracle queries, one obtains the lexicographically first satisfying assignment. Since $N = \mathrm{poly}(\ell)$, this is a polynomial-time computation with an NP oracle. □

Now, the idea is the following: if LexWit_G has small circuits on input length ℓ, then fixing N and a hard seed y inside such a circuit would leave a small circuit whose only input is the index i. That circuit would output the satisfying assignment bit a_i, contradicting Lemma 6.2.

Theorem 6.5. *Fix* $0 < \alpha \le 1$*. Assume* ETH*, and suppose that an* α*-compressed hard formula generator* $G = \{G_N\}$ *exists. Let* $q\colon \mathbb{N} \to \mathbb{R}_{>0}$ *be any polynomial-time computable function such that* $q(m) = \omega(\log m)$ *and*

$$\frac{m^{1/\alpha}}{q(m)} = \omega(1).$$

Then

$$\mathsf{P}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}\left[\frac{m^{1/\alpha}}{q(m)}\right].$$

Proof. Recall that $\ell(N) = \Theta(N^\alpha)$ is the input length of LexWit_G corresponding to formulas on N variables. Apply Lemma 6.2 with

$$s(N) = \left\lfloor \frac{N}{\sqrt{q(\ell(N))\log N}} \right\rfloor.$$

This threshold is polynomial-time computable, and the lemma applies since $q(\ell(N)) = \omega(\log N)$ implies

$$s(N)\log(s(N) + \log N) = O\left(\frac{N\log N}{\sqrt{q(\ell(N))\log N}}\right) = o(N).$$

Hence, for infinitely many N, there is a seed $y_N \in \{0,1\}^{r_\alpha(N)}$ such that $F_N = G_N(y_N)$ is satisfiable, but every satisfying assignment of F_N has circuit complexity greater than $s(N)$. Let a_N be the lexicographically first satisfying assignment of F_N.

Suppose, toward contradiction, that $\mathsf{LexWit}_G \in \mathsf{SIZE}\left[m^{1/\alpha}/q(m)\right]$. For all sufficiently large N from the infinite hard set, take a circuit for LexWit_G on inputs of length $\ell(N)$ and hardwire the bits encoding N and y_N. The resulting circuit takes only the index i as input and computes the string a_N. Its size is

$$O\left(\frac{\ell(N)^{1/\alpha}}{q(\ell(N))}\right) = O\left(\frac{N}{q(\ell(N))}\right) = o\left(\frac{N}{\sqrt{q(\ell(N))\log N}}\right) = o(s(N)).$$

For all sufficiently large hard N, this contradicts $\mathsf{CC}[a_N] > s(N)$. □

Theorem 6.6. *Fix $0 < \varepsilon < 1$. Assume* ETH, *and suppose there is a uniform family of maps*

$$G_N : \{0,1\}^{N^{1-\varepsilon}} \to \{3\text{-}CNF \text{ formulas on variables } x_1, \dots, x_N\}$$

such that $G_N(y)$ is computable in time $\mathrm{poly}(N)$, and SAT *restricted to $\mathrm{Im}(G_N)$ has no $2^{o(N)}$-time algorithm. Then, for every constant $0 < \delta < \frac{\varepsilon}{1-\varepsilon}$,*

$$\mathsf{P}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}\left[m^{1+\delta}\right].$$

Proof. Apply Theorem 6.5 with $\alpha = 1 - \varepsilon$. Let

$$\gamma \coloneqq \frac{1}{1-\varepsilon} - (1+\delta) > 0.$$

Choose the function $q(m) = m^{\gamma/2}$. Then $q(m) = \omega(\log m)$ and

$$\frac{m^{1/\alpha}}{q(m)} = m^{1+\delta+\gamma/2}.$$

Theorem 6.5 gives a language $\mathsf{LexWit}_G \in \mathsf{P}^{\mathsf{NP}}$ that is not in $\mathsf{SIZE}[m^{1+\delta+\gamma/2}]$. □

7 Open Problems

The results of this paper show that target degeneracy has a genuinely intermediate behavior: it is too rich for the known algorithms, but the available lower-bound machinery still leaves the polynomial-target regime unresolved. Theorem 3.1 shows that the dependence on $\mathrm{degen}(H)$ must be linear in the exponent whenever this parameter grows, while Theorem 4.2 shows that even degeneracy 2 can be hard when the target is quasi-polynomial. At the same time, Theorem 5.5 suggests that substantially stronger lower bounds are unlikely.

The most direct algorithmic question is whether the trivial $O^*(h^n) = 2^{O(n \log h)}$ upper bound can be improved using degeneracy. Is there an algorithm solving HOM in time $O^*\left(2^{\mathrm{poly}(\mathrm{degen}(H))\cdot n}\right)$? Such an algorithm would match Theorem 3.1 up to the dependence on $\mathrm{degen}(H)$ and would separate degeneracy from target parameters that admit $p(H)^{O(n)}$ algorithms.

The first concrete case is bounded degeneracy. For every fixed $d \geq 2$, can HOM on instances with $\mathrm{degen}(H) \leq d$ and $h \leq \mathrm{poly}(n)$ be solved in time $O^*\left(2^{O_d(n)}\right)$? Theorem 4.2 excludes this only when quasi-polynomial targets are allowed, and targets of degeneracy at most 1 are forests and hence admit plain-exponential algorithms. Thus the question is open already for $d = 2$ and polynomial-size targets.

A second direction is to understand whether the no-compression hypothesis has its own unconditional evidence. Appendix D explains why the lookup-table argument behind the barrier of Kulikov and Mihajlin [KM24] does not directly give such evidence for ETH. In short, their argument turns a claimed $2^{\Omega(s)}$ lower bound into a uniform $2^{(1-\Omega(1))N}$ algorithm for the N-variable SAT problem, by branching on all but a constant fraction of the variables and answering the remaining short queries from a precomputed table. This contradicts SETH, but it does not contradict ETH, which rules out only $2^{o(N)}$ algorithms. In the spirit of [KM24], can one nevertheless prove a barrier showing that a disproof of Hypothesis 1.3 would imply an unexpected consequence? One concrete form of this question is whether there is a generator of hard 3-SAT formulas with a linear number of bits[6].

[6] This question was raised by Daniel Lokshtanov, personal communication.

Acknowledgments

The authors used ChatGPT 5.5 to assist in strengthening the constructions in Section 3 and in checking the lower-bound calculations in Section 6. The resulting suggestions were reviewed by the authors and incorporated only after independent verification. The authors assume responsibility for all content.

A Bounded Chromatic Number

This section notes that the bounded chromatic number lower bound of [FGKM15] can be strengthened via ideas in [CFG+17].

Theorem A.1. *Unless* ETH *fails, for every constant $\alpha > 0$ there is a constant $\rho = \rho(\alpha) > 0$ such that no algorithm decides* HOM *in time*

$$O^* \left(n^{\rho \cdot n}\right)$$

on instances (G, H) satisfying $h \leq n^{\alpha}$ and $\chi(H) \leq 15$.

We state four lemmas and then combine them. We use a grouping of [CFG+17] and then use the chromatic-number reduction of [FGKM15] to get the constant bound on $\chi(H)$.

Lemma A.2 ([CFG+17, Lemma 3])**.** *For any constant d, there is a constant $\lambda = \lambda(d)$ and a polynomial-time algorithm with the following property. Given an n-vertex graph Q of maximum degree at most d and an integer r satisfying $2 \leq r \leq \sqrt{n/(2\lambda)}$, the algorithm constructs a grouping graph $\mathcal{B}$ and a labeling $\ell_{\mathcal{B}} \colon V(\mathcal{B}) \to [\lambda r]$ such that*

$$|V(\mathcal{B})| \leq n/r,$$

$\ell_{\mathcal{B}}$ is a proper coloring of $\mathcal{B}^2$, every bucket $B \in V(\mathcal{B})$ is an independent set in Q, and between any two buckets there is at most one edge of Q.

Lemma A.3 ([CFG+17, Lemma 4])**.** *Let Q be a graph and let Λ be a grouping graph of Q with label set $[L]$. Suppose that the labeling is a proper coloring of Λ^2, every bucket is an independent set in Q, and between any two buckets there is at most one edge of Q. Then one can construct, in time polynomial in the output size, a list-homomorphism instance $(\Lambda, \Gamma, \mathcal{L})$ such that*

$$Q \text{ is } 3\text{-colorable} \iff (\Lambda, \Gamma, \mathcal{L}) \text{ is satisfiable},$$

and

$$|V(\Gamma)| \leq L \cdot 4^L.$$

Lemma A.4 ([FGKM15, Lemma 4])**.** *Given a list-homomorphism instance $(\Lambda, \Gamma, \mathcal{L})$ and a proper k-coloring of Λ, one can construct in polynomial time an equivalent list-homomorphism instance $(\Lambda, \Gamma', \mathcal{L}')$ with*

$$|V(\Gamma')| = k|V(\Gamma)| \qquad \text{and} \qquad \chi(\Gamma') \leq k.$$

Lemma A.5 ([FGKM15, Lemma 5])**.** *Given a list-homomorphism instance $(\Lambda, \Gamma, \mathcal{L})$ with $|V(\Lambda)| = n$, $|V(\Gamma)| = h$, and $\chi(\Gamma) \leq t$, one can construct in polynomial time an equivalent ordinary homomorphism instance (G, H) satisfying*

$$|V(G)| \leq n + (h+1)(t+11), \qquad |V(H)| \leq (h+1)(t+11), \qquad \chi(H) \leq t + 10.$$

Corollary A.6. *There are constants $\lambda \geq 1$ and $\sigma_0 > 1$, and a polynomial-time algorithm with the following property. Given an n-vertex graph Q of maximum degree at most 4 and an integer $2 \leq r \leq \sqrt{n/(2\lambda)}$, the algorithm constructs a list-homomorphism instance $(\Lambda, \Gamma, \mathcal{L})$ such that*

$$Q \text{ is 3-colorable} \iff (\Lambda, \Gamma, \mathcal{L}) \text{ is satisfiable},$$

and

$$|V(\Lambda)| \leq \frac{5n}{r}, \qquad \chi(\Lambda) \leq 5, \qquad |V(\Gamma)| \leq \sigma_0^r.$$

Proof. Let $q \colon V(Q) \to [5]$ be a greedy proper 5-coloring of Q. Apply Lemma A.2 with $d = 4$, and let $\mathcal{B}$ be the resulting grouping graph with labeling $\ell_{\mathcal{B}} : V(\mathcal{B}) \to [L]$, where $L \leq \lambda r$.

Refine every bucket $B \in V(\mathcal{B})$ by the color classes of q: for each $a \in [5]$ with $B_a = B \cap q^{-1}(a) \neq \varnothing$, create a bucket B_a. Let Λ be the graph on these refined buckets, with two refined buckets adjacent exactly when an edge of Q runs between them, and let each refined bucket inherit the label of its parent bucket. Then $|V(\Lambda)| \leq 5|V(\mathcal{B})| \leq 5n/r$, and q gives a proper 5-coloring of Λ.

Every refined bucket is independent, and between any two refined buckets there is at most one edge of Q. The inherited labeling remains a proper coloring of Λ^2: different parent buckets inherit this from $\mathcal{B}^2$, while two buckets with the same parent are not adjacent and cannot have a common neighbor without creating two edges between one pair of parent buckets.

Apply Lemma A.3 to the refined grouping Λ. It gives an equivalent list-homomorphism instance $(\Lambda, \Gamma, \mathcal{L})$ with

$$|V(\Gamma)| \leq L \cdot 4^L \leq \sigma_0^r$$

for a constant σ_0 depending only on λ. □

Proof of Theorem A.1. Fix $\alpha > 0$, and let $\beta > 0$ be the bounded-degree 3-Coloring constant from Theorem 2.4. Let λ and σ_0 be the constants from Corollary A.6.

Choose a constant $\sigma \geq \max\{\sigma_0, 2\}$ large enough that the following consequence holds for every $r \geq 2$. If Corollary A.6 is followed by Lemma A.4 with $k = 5$ and then by Lemma A.5 with $t = 5$, the resulting ordinary homomorphism instance (G_0, H_0) is equivalent to 3-coloring of Q and satisfies

$$|V(G_0)| \leq \frac{5n}{r} + \sigma^r, \qquad |V(H_0)| \leq \sigma^r, \qquad \chi(H_0) \leq 15.$$

Indeed, the target recoloring changes the list target size by a factor 5, and list removal adds only $(h+1)(5+11)$ vertices to each side.

Set

$$\delta = \frac{\min\{1, \alpha\}}{4 \log \sigma}.$$

Given an n-vertex maximum-degree-4 graph Q, put $r = \lfloor \delta \log n \rfloor$. For all sufficiently large n, the condition $2 \leq r \leq \sqrt{n/(2\lambda)}$ holds, so Lemma A.6 applies.

For the resulting ordinary homomorphism instance,

$$\sigma^r \leq \sigma^{\delta \log n} = n^{\delta \log \sigma} \leq n^{1/4}.$$

Therefore, for all sufficiently large n,

$$|V(G_0)| \leq \frac{5n}{r} + n^{1/4} \leq \frac{6n}{r}.$$

Pad G_0 with isolated vertices so that the final source graph G has exactly

$$N = \left\lceil \frac{6n}{r} \right\rceil$$

vertices, and set $H := H_0$. Adding isolated vertices does not affect the existence of a homomorphism, so

$$Q \text{ is 3-colorable} \iff G \to H.$$

The chromatic-number bound remains $\chi(H) \leq 15$.

The target size is polynomial in the final source size. Indeed,

$$|V(H)| \leq \sigma^r \leq n^{\alpha/4}.$$

Since $N = \Theta(n/\log n)$, we have $n^{\alpha/4} \leq N^\alpha$ for all sufficiently large n. Thus the constructed instances satisfy

$$|V(G)| = N, \qquad |V(H)| \leq N^\alpha, \qquad \chi(H) \leq 15.$$

Now suppose that, for some constants $\rho > 0$ and $d \geq 0$, this restricted class of HOM instances can be decided in time

$$O(N^{\rho N} \cdot |V(H)|^d).$$

For all sufficiently large n, we have $r \geq (\delta/2)\log n$, and hence

$$N \leq \frac{7n}{r} \leq \frac{14n}{\delta \log n}.$$

Since also $\log N \leq \log n$, it follows that

$$N \log N \leq \frac{14}{\delta} n.$$

As $|V(H)| \leq N^\alpha$, the assumed algorithm would solve 3-Coloring in time

$$2^{\rho N \log N} N^{O(1)} \leq 2^{(14\rho/\delta + o(1))n}.$$

Choose $\rho = \beta\delta/28$. Then the last running time is $2^{(\beta/2+o(1))n}$, contradicting Theorem 2.4. □

B Graph Parameters

This appendix defines all target-side graph parameters from Figure 1 and proves the implications represented by its arrows.

- The chromatic number $\chi(G)$ is the least number of colors in a proper vertex coloring of G.
- The maximum degree $\Delta(G)$ is the maximum degree of a vertex of G.
- The degeneracy $\text{degen}(G)$ is the least integer d such that every non-empty subgraph of G has a vertex of degree at most d.

- A tree decomposition of G is a tree T with bags $\beta(x) \subseteq V(G)$, $x \in V(T)$, such that every vertex and every edge of G is covered by a bag and the bags containing any fixed vertex form a non-empty connected subtree of T. The treewidth $\mathrm{tw}(G)$ is the minimum value of $\max_{x\in V(T)} |\beta(x)| - 1$ over all tree decompositions of G [RS86].
- For $k \geq 1$, a k-expression builds k-labeled graphs by creating one vertex of a chosen label, taking disjoint unions, joining all vertices of one label to all vertices of another label, and relabeling vertices. The clique-width $\mathrm{cw}(G)$ is the least k such that some k-expression represents G after labels are forgotten [CER93, CO00].
- The extended clique-width $\mathrm{ecw}(G)$ is the analogous minimum for extended k-expressions of Bulatov and Dadsetan [BD20]. Such expressions add to k-expressions the connect operation $\eta_{\mathcal{T}}$ between two labeled graphs and the beta operation $\beta_{\vec{n},\sigma,\mathcal{S}}$, which replaces each vertex of label i by $n_i + 1$ labeled copies and keeps edges according to $\mathcal{S}$.
- For a linear order $v_1, \ldots, v_m$ of $V(G)$, its bandwidth is $\max\{|i - j| : v_iv_j \in E(G)\}$. The bandwidth $\mathrm{bw}(G)$ is the minimum such value over all linear orders, and the complement bandwidth is $\mathrm{bw}(\overline{G})$ [Har64].
- A track layout of G is a proper coloring γ together with a linear order $\prec$ of $V(G)$ such that no two edges u_1u_2, v_1v_2 with $\gamma(u_1) = \gamma(v_1)$ and $\gamma(u_2) = \gamma(v_2)$ satisfy $u_1 \prec v_1$ and $v_2 \prec u_2$. The track number $\mathrm{tn}(G)$ is the least number of colors in a track layout [DPW04].
- Given a proper coloring $\gamma\colon V(G) \to [k]$, let $\mathbf{\Gamma}_{G,\gamma}$ be the binary constraint language on domain $V(G)$ with relations
$$R_{ij} = \{(u, v) \in V(G)^2 \colon \gamma(u) = i,\ \gamma(v) = j,\ uv \in E(G)\}, \qquad i, j \in [k].$$
A ternary operation $m\colon D^3 \to D$ is a majority operation if $m(b, a, a) = m(a, b, a) = m(a, a, b) = a$ for all $a, b \in D$. For tuples t_1, t_2, t_3 of the same arity, write $f(t_1, t_2, t_3)$ for the tuple obtained by applying f coordinatewise. A persistent majority triple for a binary crisp constraint language $\mathbf{\Gamma}$ over domain D is a triple (f_1, f_2, f_3) of operations $D^3 \to D$ such that f_1 is majority,
$$\{\{f_1(a_1, a_2, a_3), f_2(a_1, a_2, a_3), f_3(a_1, a_2, a_3)\}\} = \{\{a_1, a_2, a_3\}\}$$
for all $a_1, a_2, a_3 \in D$, and
$$\{\{f_1(t_1, t_2, t_3), f_2(t_1, t_2, t_3), f_3(t_1, t_2, t_3)\}\} = \{\{t_1, t_2, t_3\}\}$$
for every relation $R \in \mathbf{\Gamma}$ and all $t_1, t_2, t_3 \in R$. A persistent majority coloring of G is a proper coloring γ such that $\mathbf{\Gamma}_{G,\gamma}$ has a persistent majority triple. The persistent majority number $\mathrm{pmn}(G)$ is the least number of colors in a persistent majority coloring [Car26].

B.1 Implications in the Parameter Map

Lemma B.1. *For every finite graph G, the following implications hold:*
$$\begin{aligned} \mathrm{cw}(G) &\leq 2^{\mathrm{tw}(G)+2}, & \mathrm{cw}(G) &\leq 2^{\mathrm{bw}(\overline{G})+3}, \\ \mathrm{ecw}(G) &\leq \mathrm{cw}(G), & \mathrm{tn}(G) &\leq f_{\mathrm{tw}}(\mathrm{tw}(G)), \\ \mathrm{pmn}(G) &\leq \mathrm{tn}(G), & \mathrm{pmn}(G) &\leq \Delta(G)^2 + 1, \end{aligned}$$
for some function f_{tw}.

Proof. $\mathrm{cw}(G) \le 2^{\mathrm{tw}(G)+2}$ is due to Courcelle and Olariu [CO00]. Every k-expression is an extended k-expression, hence $\mathrm{ecw}(G) \le \mathrm{cw}(G)$.

Let $b = \mathrm{bw}(\overline{G})$. Choose an ordering $v_1, \dots, v_m$ of $V(G)$ witnessing bandwidth at most b in $\overline{G}$. Then the bags $B_i = \{v_j : i \le j \le i + b\}$ form a path decomposition of $\overline{G}$ of width at most b, because every edge of $\overline{G}$ has both endpoints in one such interval. Therefore $\mathrm{tw}(\overline{G}) \le b$, and the previous paragraph gives $\mathrm{cw}(\overline{G}) \le 2^{b+2}$. Clique-width is bounded under complementation, more precisely $\mathrm{cw}(\overline{F}) \le 2\,\mathrm{cw}(F)$ for every graph F [CO00]. Applying this to $F = \overline{G}$ yields $\mathrm{cw}(G) \le 2^{b+3}$, and then $\mathrm{ecw}(G) \le 2^{b+3}$.

Graphs of bounded treewidth have bounded track number due to [DMW05]. On the other hand, bounded track number or bounded maximum degree imply bounded persistent majority number due to [Car26]. ☐

Lemma B.2. *For every graph G,*

$$\mathrm{degen}(G) \le 4(\mathrm{pmn}(G) - 1).$$

Proof. Let $k = \mathrm{pmn}(G)$, and let $\gamma\colon V(G) \to [k]$ be a persistent majority coloring witnessed by a persistent majority triple (f_1, f_2, f_3). For colors $i < j$, let B_{ij} be the bipartite graph induced by the edges between color classes $\gamma^{-1}(i)$ and $\gamma^{-1}(j)$.

We first show that each B_{ij} is 2-degenerate. Suppose to the contrary that some subgraph B of B_{ij} has minimum degree at least 3. Choose a vertex x on the i-side of B with three distinct neighbors y_1, y_2, y_3 on the j-side. For each $r \in \{1, 2, 3\}$, choose a neighbor $x_r \neq x$ of y_r in B.

Apply f_1 to the three tuples $(x, y_1), (x, y_2), (x_3, y_3) \in R_{ij}$. On the first coordinate, $f_1(x, x, x_3) = x$. Since the output tuple $f_1((x, y_1), (x, y_2), (x_3, y_3))$ is one of the three input tuples, it must be (x, y_1) or (x, y_2). Hence $f_1(y_1, y_2, y_3) \in \{y_1, y_2\}$. The same argument applied to $(x, y_1), (x_2, y_2), (x, y_3)$ gives $f_1(y_1, y_2, y_3) \in \{y_1, y_3\}$, and applied to $(x_1, y_1), (x, y_2), (x, y_3)$ gives $f_1(y_1, y_2, y_3) \in \{y_2, y_3\}$. These three sets have empty intersection, a contradiction. Thus each B_{ij} is 2-degenerate.

Now let $U \subseteq V(G)$ and set $U_i = U \cap \gamma^{-1}(i)$. For each pair $i < j$, the induced bipartite graph between U_i and U_j is 2-degenerate, and therefore has at most $2(|U_i| + |U_j|)$ edges. Because γ is proper, all edges of $G[U]$ lie between distinct color classes, so

$$|E(G[U])| \le \sum_{1 \le i < j \le k} 2(|U_i| + |U_j|) = 2(k-1)|U|.$$

Every induced subgraph of G has average degree at most $4(k-1)$. Therefore every subgraph of G has a vertex of degree at most $4(k-1)$, as required. ☐

B.2 Padding Lower Bounds

We also note that the upper bounds in Figure 1 are tight under ETH for all parameters (except complement bandwidth).

Observation B.3. *Let μ be a graph parameter such that $\mu(H) \le |V(H)|$ for every graph H, and suppose that μ is unchanged by adding isolated vertices. Let $p = p(n)$ and $h = h(n)$ be integer-valued functions with $3 \le p(n) \le h(n)$. Unless* ETH *fails, there is no algorithm that solves* HOM *on all instances with $|V(G)| = n$, $|V(H)| = h(n)$, and $\mu(H) \le p(n)$ in time $p(n)^{o(n)}$.*

Proof. By the lower bound of [CFG⁺17], under ETH, HOM on instances with $|V(G)| = n$ and $|V(H_0)| = p(n)$ has no $p(n)^{o(n)}$-time algorithm. Given such an instance (G, H_0), add $h(n) - p(n)$ isolated vertices to H_0 and call the resulting target H. Then $|V(H)| = h(n)$ and, by the assumption on μ,

$$\mu(H) = \mu(H_0) \leq |V(H_0)| = p(n).$$

Moreover, $G \to H$ if and only if $G \to H_0$. Indeed, dummy target vertices are isolated, so any source vertex mapped to one of them is isolated in G and can be remapped to an arbitrary vertex of H_0. Thus a $p(n)^{o(n)}$-time algorithm for the padded instances would give a $p(n)^{o(n)}$-time algorithm for the hard instances of [CFG⁺17], a contradiction. □

Complement bandwidth $\mathrm{bw}(\overline{H})$ increases when adding isolated vertices, so the observation above does not follow. Moreover, for any graph H on h vertices with $\mathrm{bw}(\overline{H}) \leq p$ it follows that $\omega(H) \geq \left\lceil \frac{h}{p+1} \right\rceil$, hence if $h(n) > (n-1)(p(n)+1)$, then for every graph G on n vertices, there is a homomorphism $G \to H$. Thus, the above observation is false for complement bandwidth.

C The Graph Homomorphism Problem and Its Complexity Dichotomies

The lower bound $2^{\Omega(n \log h)}$ for HOM holds for the general case when both graphs G and H are part of the input and are not restricted in any way [CFG⁺17]. At the same time, in various applications of the Graph Homomorphism problem, one or both of these graphs may come from a restricted graph family or even be fixed. There is a rich variety of results studying the complexity of special cases of HOM. Complexity dichotomies ask where the boundary lies between the tractable and intractable sides of the relevant complexity model, such as P versus NP-hardness, FPT versus W[1]-hardness, or FP versus #P-hardness. Fine-grained classifications ask, once the problem is intractable, which structural features determine the correct exponential running time.

Fixed and Restricted Source Graphs. For the *fixed source* problem $\mathsf{HOM}(G, \cdot)$, the brute-force algorithm with running time $O(h^n)$ is polynomial. For *restricted source* graph classes $\mathcal{G}$, the question is what properties of $\mathcal{G}$ make $\mathsf{HOM}(\mathcal{G}, \cdot)$ tractable. Dalmau, Kolaitis, and Vardi [DKV02] and Grohe [Gro07] showed that $\mathsf{HOM}(\mathcal{G}, \cdot)$, parametrized by the size of the source graph, is in FPT precisely when the cores of graphs in $\mathcal{G}$ have bounded treewidth, and is W[1]-hard otherwise. See Figure 6. Dalmau and Jonsson [DJ04] proved the analogous result for counting homomorphisms.

$\mathrm{tw}(\mathrm{core}(\mathcal{G})) = O(1)$	$\mathrm{tw}(\mathrm{core}(\mathcal{G})) = \omega(1)$
easy: $\mathsf{HOM}(\mathcal{G}, \cdot) \in \mathsf{FPT}$	hard: $\mathsf{HOM}(\mathcal{G}, \cdot)$ is W[1]-hard

Figure 6: Parameterized complexity dichotomy for the source-restricted problem $\mathsf{HOM}(\mathcal{G}, \cdot)$.

Fine-grained source-side results further show that the known dynamic-programming exponents are often the right ones. For vertex cover, [FGKM15] give an $O(h^{\mathrm{vc}(G)})$ algorithm and a matching ETH lower bound $h^{\Omega(\mathrm{vc}(G))}$. Marx [Mar10] showed that the $h^{O(\mathrm{tw}(G))}$ algorithm cannot be significantly improved under ETH. Okrasa and Rzążewski [OR21] showed that if H is a projective core, then there is no $(h-\varepsilon)^{\mathrm{tw}(G)}n^{O(1)}$ algorithm, for any $\varepsilon > 0$, under SETH. Since almost all graphs are projective cores [HN92, ŁN04], this covers almost all targets; moreover, the result can be lifted to all targets under long-standing conjectures on the properties of projective cores [Lar02, LT01]. This was extended by [GHK$^+$24], who proved both lower and upper bounds $O^*(s(H)^{\mathrm{cw}(G)})$ under the same conjectures, where s is a signature number of H. Okrasa, Piecyk, and Rzążewski [OPR20], building on the result of Egri, Marx, and Rzążewski [EMR18], showed that for all graphs H where List Homomorphism (LHOM) is NP-hard, it also cannot be solved in time $(h-\varepsilon)^{\mathrm{tw}(G)}n^{O(1)}$, for any $\varepsilon > 0$, under SETH. For counting list homomorphisms, Focke, Marx, and Rzążewski [FMR24] identify the exact exponent base in running time and prove matching #SETH lower bounds. [GMN$^+$24] provide a characterization of the case when G is parametrized by cutwidth in terms of the asymptotic rank parameter they introduce.

Fixed Target Graphs. The *fixed target* version HOM$(\cdot, H)$ is also well understood. Hell and Nešetřil [HN90] proved that HOM$(\cdot, H)$ is polynomial-time solvable when H is bipartite, and is NP-complete otherwise; Meyer and Opršal recently gave a topological proof [MO25]. Thus, the coarse-grained complexity class of HOM$(\cdot, H)$ is determined by an easy-to-compute parameter of H: whether $\chi(H) \le 2$ or not; see Figure 7. The Feder–Vardi dichotomy conjecture for finite-domain CSPs [FV98] was proved independently by Bulatov and Zhuk, giving a complete algebraic classification of fixed-target tractability [Bul17, Zhu17].

$\chi(H) \le 2$	$\chi(H) > 2$
easy: HOM$(\cdot, H)$ $\in$ P	hard: HOM$(\cdot, H)$ is NP-hard

Figure 7: Complexity dichotomy for the fixed-target problem HOM$(\cdot, H)$.

The same dichotomy philosophy extends to several homomorphism variants. Feder and Hell [FH98] showed that if each vertex of H has a loop, then LHOM$(\cdot, H)$ is in P if H is an interval graph and NP-complete otherwise. Feder, Hell, and Huang [FHH99] showed that if H has no loops, then LHOM$(\cdot, H)$ is in P if the complement of H is a circular-arc graph with clique-cover number two, and is NP-complete otherwise. Finally, Feder, Hell, and Huang [FHH03] defined a new class of graphs with possible loops, called bi-arc graphs, and showed that if H is a bi-arc graph, then LHOM$(\cdot, H)$ can be solved in polynomial time, and otherwise the problem is NP-complete. In the reflexive case, bi-arc graphs coincide with interval graphs; in the irreflexive case, they coincide with bipartite graphs whose complement is a circular-arc graph. Dyer and Greenhill [DG00] showed that #HOM$(\cdot, H)$ is #P-complete unless every connected component of H is an isolated vertex without a loop, a complete graph with all loops present, or a complete unlooped bipartite graph; otherwise it is in FP. This dichotomy was extended in numerous ways: to weighted graphs [BG05, GGJT10, GT11, CCL13, GCD23], directed graphs [DGP07, CC17], and the number of homomorphisms modulo a fixed prime [FJ15, GGR14, GGR16]. Chen, Curticapean, and Dell [CCD19] showed that the hard cases from [DG00] require $2^{\Omega(n)}$ time under ETH.

For finite-domain counting CSPs, the exact tractable side is also sharply classified by algebraic conditions [Bul13, DR13, FGŽ19]. We refer to Bulatov's survey [Bul18] and the book of Hell and Nešetřil [HN26] for a broader background.

Restricted Target Graphs. The fine-grained picture becomes much less complete for the *restricted target* version $\mathsf{HOM}(\cdot, \mathcal{H})$. Several recent results identify properties of the target class $\mathcal{H}$ that bring $\mathsf{HOM}(\cdot, \mathcal{H})$ down to single-exponential time. Fomin, Heggernes, and Kratsch [FHK07] gave an $O^*((2\,\mathrm{tw}(H)+1)^n)$ algorithm. Wahlström [Wah11] gave an $O^*((2\,\mathrm{cw}(H)+1)^{\max\{n,h\}})$ algorithm for bounded clique-width. Bulatov and Dadsetan [BD20] extended this to bounded extended clique-width, giving an $O^*((4\,\mathrm{ecw}+4)^h + (2\,\mathrm{ecw}+1)^n)$ algorithm. Fomin, Golovnev, Kulikov, and Mihajlin [FGKM15] gave an $O^*(\Delta(H)^n)$ algorithm for bounded maximum degree, while Rzążewski [Rzą14] gave an $O^*((\mathrm{bw}(\overline{H})+2)^n)$ algorithm for bounded complement bandwidth. Recently, Carbonnel [Car26] proved a $2^{O(n)}$ upper bound when $\mathcal{H}$ excludes a fixed topological minor, and also gave $O^*(\mathrm{pmn}(H)^{n+h})$ and $O^*(\mathrm{tn}(H)^{n+h})$ algorithms. On the other hand, bounded chromatic number is not enough: [FGKM15] rules out $2^{O(n)}$ algorithms for this case unless ETH fails.

The Fine-Grained Dichotomy for Homomorphism. A natural target-side dream is to find a structural graph parameter p that describes the plain-exponential frontier for $\mathsf{HOM}(\cdot, \mathcal{H})$. Known algorithms place parameters such as bounded treewidth and excluded topological minors on the algorithmic side, while bounded chromatic number is far from sufficient. The weakest useful form of such a dichotomy would separate target families admitting a $2^{O(n)}$ algorithm from those that require super-single-exponential time. This raises the following question:

> Is there a natural graph parameter p such that bounded $p(\mathcal{H})$ implies that $\mathsf{HOM}(\cdot, \mathcal{H}) \in 2^{O(n)}$ when $h \le n$, whereas unbounded $p(\mathcal{H})$ implies that there is no such algorithm under ETH?

Such fine-grained dichotomies are known in several settings. For example, Bringmann, Fischer, and Künnemann [BFK19] introduced a hardness parameter H for every $\exists^k\forall$ graph property φ, and showed that the baseline $O(m^k)$ algorithm for deciding whether a graph on m vertices satisfies φ can be improved if and only if $H(\varphi) \le 2$, assuming the Hyper-Clique Hypothesis. Berkholz, Keppeler, and Schweikardt [BKS17] established a dichotomy for dynamic conjunctive-query answering, showing that it can be computed with linear preprocessing time and constant update time if and only if the homomorphic core of the query is q-hierarchical. Otherwise, the size of the query result cannot be maintained with sublinear update time. Moreover, Brand, Dell, and Roth [BDR16] established a dichotomy for the problem $T_{x,y}$, where $(x, y) \in \mathbb{Q}^2$ is a fixed rational point. Given a graph G, the task is to evaluate the Tutte polynomial [Tut54] at the point (x, y), that is, to compute $T_{x,y}(G) = T(G; x, y)$. This problem belongs to FP if and only if $(x-1)(y-1) = 1$. For every other point, it is $\#\mathsf{P}$-hard and cannot be solved in time $2^{o(n)}$ under $\#\mathsf{ETH}$ [JVW90, DHM$^+$14, HT10, Cur18, BDR16]. The same work [BDR16] also established a dichotomy for the problem of counting satisfying assignments of Boolean CSP instances over a constraint language Γ. The problem belongs to FP if and only if every relation in Γ is affine over $\mathsf{GF}(2)$. Otherwise, it is $\#\mathsf{P}$-complete and cannot be solved in time $2^{o(n)}$ under $\#\mathsf{ETH}$ [CH96, BDR16].

C.1 Obstruction for Monotone Parameters

We finish with a simple obstruction that shows that any monotone parameter cannot establish the dichotomy.

Theorem C.1. *Let p be an unbounded graph parameter such that*

$$A \leq_{\mathsf{ind}} H \qquad \Longrightarrow \qquad p(A) \leq p(H),$$

where $A \leq_{\mathsf{ind}} H$ means that A is an induced subgraph of H. Then there is a target family $\mathcal{H}$ with $\sup_{H\in\mathcal{H}} p(H) = \infty$ such that $\mathsf{HOM}(\cdot, \mathcal{H})$ on inputs (G, H) with $|V(G)| = n$, $H \in \mathcal{H}$, and $|V(H)| \leq n$ is solvable in time $2^{O(n)}$.

The same conclusion holds if p is monotone under subgraphs or under minors.

Proof. Since p is unbounded, choose graphs $A_1, A_2, \ldots$ such that $p(A_j) \geq j$ for all j. By taking a subsequence of it we may assume that the sizes $a_j = |V(A_j)|$ are strictly increasing.

For every integer $i \geq 1$, let $j(i)$ be the largest index j such that $a_j \leq i/2$, if such an index exists. If no such index exists, set $H_i = K_i$. If $j(i)$ exists, set

$$H_i = K_{i-a_{j(i)}} \sqcup A_{j(i)}.$$

Let $\mathcal{H} = \{H_i : i \geq 1\}$. Then $|V(H_i)| = i$. In the nontrivial case, $A_{j(i)}$ is an induced subgraph of H_i, so monotonicity gives

$$p(H_i) \geq p(A_{j(i)}) \geq j(i).$$

As $i \to \infty$, also $j(i) \to \infty$, and hence $\sup_{H\in\mathcal{H}} p(H) = \infty$.

It remains to observe that every H_i is homomorphically equivalent to a clique. This is clear when $H_i = K_i$. In the nontrivial case, put $j = j(i)$ and $k_i = i - a_j$. Since $a_j \leq i/2$, we have $k_i \geq a_j$. The clique component gives an inclusion $K_{k_i} \to H_i$. Conversely, map the clique component of H_i identically to K_{k_i} and map $V(A_j)$ injectively into $V(K_{k_i})$. This is a homomorphism because K_{k_i} is complete and there are no edges between the two components of H_i. Thus H_i and K_{k_i} are homomorphically equivalent, and for every graph G,

$$G \to H_i \qquad \Longleftrightarrow \qquad G \to K_{k_i}.$$

Now consider an input (G, H_i) with $|V(G)| = n$ and $i = |V(H_i)| \leq n$. From H_i one can find, in polynomial time, a complete connected component C of size $\ell_i \geq i/2$. Then K_{ℓ_i} embeds into H_i, and every other connected component has at most $i - \ell_i \leq \ell_i$ vertices, so it maps injectively to K_{ℓ_i}. Thus H_i is homomorphically equivalent to K_{ℓ_i}, and the problem is exactly ℓ_i-colorability of G. Since $\ell_i \leq i \leq n$, it can be decided in time $2^n n^{O(1)}$ [BHK09]. Thus $\mathsf{HOM}(\cdot, \mathcal{H})$ in the regime $h \leq n$ is solvable in time $2^{O(n)}$.

The construction also proves the subgraph- and minor-monotone variants, since $A_{j(i)}$ is not only an induced subgraph of H_i, but also a subgraph and a minor of H_i. ☐

Consequently, parameters such as $\mathrm{tw}(H)$, $\mathrm{degen}(H)$, $\Delta(H)$, $\chi(H)$, and $\mathrm{cw}(H)$ cannot give the hard direction of the family-level question.

D SETH Version of the No-Compression Barrier

For the sake of completeness, we outline the main idea behind the result of Kulikov and Mihajlin [KM24].

Lemma D.1. *Let $\mathcal{A}$ be a decision problem whose instances are encoded by s bits, and assume that $\mathcal{A}$ can be solved in time $2^{O(s)}$.*

Then, if a fine-grained reduction from SAT *to $\mathcal{A}$ establishes, under* SETH*, a lower bound $2^{\Omega(s)}$ for $\mathcal{A}$, then* SETH *is false.*

Proof. Write the claimed lower bound as $2^{\gamma s}$ for a constant $\gamma > 0$. The fine-grained reduction gives constants $\kappa, \delta > 0$ such that, on formulas with N variables, all oracle queries have length at most κN, and the reduction with a $2^{(1-\varepsilon)\gamma s}$-time oracle runs in time at most $2^{(1-\delta)N}$ for some fixed $\varepsilon > 0$.

Choose a small constant $\beta > 0$. Given an n-variable SAT instance, precompute the answers to all $\mathcal{A}$-instances of length at most $\kappa\beta n$ using the $2^{O(s)}$ algorithm. For β small enough this table takes time at most $2^{(1-\sigma)n}$ for some $\sigma > 0$. After that, branch on all but βn variables. For each of the resulting $2^{(1-\beta)n}$ formulas, run the assumed reduction. Its oracle queries have length at most $\kappa\beta n$, so they are answered from the table. The total time is

$$2^{(1-\sigma)n} + 2^{(1-\beta)n} \cdot 2^{(1-\delta)\beta n} = 2^{(1-\Omega(1))n},$$

which contradicts SETH. $\square$

References

[AG09] Noga Alon and Shai Gutner. Linear Time Algorithms for Finding a Dominating Set of Fixed Size in Degenerated Graphs. *Algorithmica*, 54(4):544–556, 2009. 4

[Akk73] Eralp A. Akkoyunlu. The Enumeration of Maximal Cliques of Large Graphs. *SIAM J. Comput.*, 2(1):1–6, 1973. 4

[BD20] Andrei A. Bulatov and Amineh Dadsetan. Counting Homomorphisms in Plain Exponential Time. In *ICALP*, LIPIcs, pages 21:1–21:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 3, 4, 40, 44

[BDR16] Cornelius Brand, Holger Dell, and Marc Roth. Fine-Grained Dichotomies for the Tutte Plane and Boolean #CSP. In *IPEC*, volume 63 of *LIPIcs*, pages 9:1–9:14, 2016. 44

[BFK19] Karl Bringmann, Nick Fischer, and Marvin Künnemann. A Fine-Grained Analogue of Schaefer's Theorem in P: Dichotomy of $\exists^k\forall$-Quantified First-Order Graph Properties. In *CCC*, volume 137 of *LIPIcs*, pages 31:1–31:27, 2019. 44

[BG05] Andrei A. Bulatov and Martin Grohe. The Complexity of Partition Functions. *Theor. Comput. Sci.*, 348:148–186, 2005. 43

[BGL+22] Suman K. Bera, Lior Gishboliner, Yevgeny Levanzov, C. Seshadhri, and Asaf Shapira. Counting Subgraphs in Degenerate Graphs. *J. ACM*, 69(3):23:1–23:21, 2022. 4

[BHK09] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set Partitioning via Inclusion-Exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009. 3, 45

[BKS17] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. Answering Conjunctive Queries under Updates. In *PODS*, pages 303–318. ACM, 2017. 44

[BLR23] Marco Bressan, Matthias Lanzinger, and Marc Roth. The Complexity of Pattern Counting in Directed Graphs, Parameterised by the Outdegree. In *STOC*, pages 542–552, 2023. 4

[BPS20] Suman K. Bera, Noujan Pashanasangi, and C. Seshadhri. Linear Time Subgraph Counting, Graph Degeneracy, and the Chasm at Size Six. In *ITCS*, LIPIcs, pages 38:1–38:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 4

[BPS21] Suman K. Bera, Noujan Pashanasangi, and C. Seshadhri. Near-Linear Time Homomorphism Counting in Bounded Degeneracy Graphs: The Barrier of Long Induced Cycles. In *SODA*, pages 2315–2332, 2021. 4

[BR21] Marco Bressan and Marc Roth. Exact and Approximate Pattern Counting in Degenerate Graphs: New Algorithms, Hardness Results, and Complexity Dichotomies. In *FOCS*, pages 276–285. IEEE, 2021. 4

[Bre21] Marco Bressan. Faster Algorithms for Counting Subgraphs in Sparse Graphs. *Algorithmica*, 83(8):2578–2605, 2021. 4

[Bul13] Andrei A. Bulatov. The Complexity of the Counting Constraint Satisfaction Problem. *J. ACM*, 60:34:1–34:41, 2013. 44

[Bul17] Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *FOCS*, pages 319–330, 2017. 43

[Bul18] Andrei A. Bulatov. Constraint Satisfaction Problems: Complexity and Algorithms. *ACM SIGLOG News*, 5:4–24, 2018. 44

[Car26] Clément Carbonnel. Faster Algorithms for Graph Homomorphism via Tractable Constraint Satisfaction. *CoRR*, abs/2602.23000, 2026. 3, 4, 40, 41, 44

[CC17] Jin-Yi Cai and Xi Chen. Complexity of Counting CSP with Complex Weights. *J. ACM*, 64:19:1–19:39, 2017. 43

[CCC06] Leizhen Cai, Siu Man Chan, and Siu On Chan. Random Separation: A New Method for Solving Fixed-Cardinality Optimization Problems. In *IWPEC*, Lecture Notes in Computer Science, pages 239–250. Springer, 2006. 4

[CCD19] Hubie Chen, Radu Curticapean, and Holger Dell. The Exponential-Time Complexity of Counting (Quantum) Graph Homomorphisms. In *Graph-Theoretic Concepts in Computer Science*, pages 364–378, 2019. 43

[CCL13] Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph Homomorphisms with Complex Values: A Dichotomy Theorem. *SIAM J. Comput.*, 42:924–1029, 2013. 43

[CER93] Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-Rewriting Hypergraph Grammars. *J. Comput. Syst. Sci.*, 46(2):218–270, 1993. 40

[CFG+17] Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socała. Tight Lower Bounds on Graph Embedding Problems. *J. ACM*, 64:18:1–18:22, 2017. 3, 6, 37, 42

[CGH17] Marek Cygan, Fabrizio Grandoni, and Danny Hermelin. Tight Kernel Bounds for Problems on Graphs with Small Degeneracy. *ACM Trans. Algorithms*, 13(3):43:1–43:22, 2017. 4

[CH96] Nadia Creignou and Miki Hermann. Complexity of Generalized Satisfiability Counting Problems. *Inf. Comput.*, 125(1):1–12, 1996. 44

[CHKX06] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong Computational Lower Bounds via Parameterized Complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. 3

[CM16] Radu Curticapean and Dániel Marx. Tight Conditional Lower Bounds for Counting Perfect Matchings on Graphs of Bounded Treewidth, Cliquewidth, and Genus. In *SODA*, pages 1650–1669, 2016. 8

[CN85] Norishige Chiba and Takao Nishizeki. Arboricity and Subgraph Listing Algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985. 4

[CO00] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discret. Appl. Math.*, 101(1-3):77–114, 2000. 40, 41

[CPPW12] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Kernelization hardness of connectivity problems in d-degenerate graphs. *Discret. Appl. Math.*, 160(15):2131–2141, 2012. 4

[Cur18] Radu Curticapean. Block interpolation: A framework for tight exponential-time counting complexity. *Inf. Comput.*, 261:265–280, 2018. 44

[DG00] Martin E. Dyer and Catherine S. Greenhill. The complexity of counting graph homomorphisms. *Random Struct. Algorithms*, 17(3-4):260–289, 2000. 43

[DGP07] Martin E. Dyer, Leslie Ann Goldberg, and Mike Paterson. On Counting Homomorphisms to Directed Acyclic Graphs. *J. ACM*, 54:27, 2007. 43

[DHM+14] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. Exponential Time Complexity of the Permanent and the Tutte Polynomial. *ACM Trans. Algorithms*, 10:21:1–21:32, 2014. 8, 44

[DJ04] Víctor Dalmau and Peter Jonsson. The Complexity of Counting Homomorphisms Seen from the Other Side. *Theor. Comput. Sci.*, 329:315–323, 2004. 42

[DKV02] Víctor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *Principles and Practice of Constraint Programming*, pages 310–326, 2002. 42

[DM12] Holger Dell and Dániel Marx. Kernelization of packing problems. In *SODA*, pages 68–81. SIAM, 2012. 4

[DMW05] Vida Dujmović, Pat Morin, and David R. Wood. Layout of Graphs with Bounded Tree-Width. *SIAM J. Comput.*, 34(3):553–579, 2005. 41

[DPW04] Vida Dujmović, Attila Pór, and David R. Wood. Track Layouts of Graphs. *Discrete Math. Theor. Comput. Sci.*, 6:497–521, 2004. 40

[DR13] Martin E. Dyer and David Richerby. An Effective Dichotomy for the Counting Constraint Satisfaction Problem. *SIAM J. Comput.*, 42:1245–1274, 2013. 44

[Dvo10] Zdeněk Dvořák. On Recognizing Graphs by Numbers of Homomorphisms. *J. Graph Theory*, 64(4):330–342, 2010. 4

[ELS10] David Eppstein, Maarten Löffler, and Darren Strash. Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time. In *ISAAC (1)*, Lecture Notes in Computer Science, pages 403–414. Springer, 2010. 4

[EMR18] László Egri, Dániel Marx, and Paweł Rzążewski. Finding List Homomorphisms from Bounded-Treewidth Graphs to Reflexive Graphs: A Complete Complexity Characterization. In *STACS*, pages 27:1–27:15, 2018. 43

[FGKM15] Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, and Ivan Mihajlin. Lower Bounds for the Graph Homomorphism Problem. In *ICALP (1)*, pages 481–493, 2015. 3, 4, 6, 23, 37, 42, 44

[FGŽ19] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The Complexity of Counting Surjective Homomorphisms and Compactions. *SIAM J. Discret. Math.*, 33:1006–1043, 2019. 44

[FH98] Tomás Feder and Pavol Hell. List Homomorphisms to Reflexive Graphs. *J. Comb. Theory B*, 72(2):236–250, 1998. 43

[FHH99] Tomás Feder, Pavol Hell, and Jing Huang. List Homomorphisms and Circular Arc Graphs. *Comb.*, 19(4):487–505, 1999. 43

[FHH03] Tomás Feder, Pavol Hell, and Jing Huang. Bi-Arc Graphs and the Complexity of List Homomorphisms. *J. Graph Theory*, 42:61–80, 2003. 43

[FHK07] Fedor V. Fomin, Pinar Heggernes, and Dieter Kratsch. Exact Algorithms for Graph Homomorphisms. *Theory Comput. Syst.*, 41:381–393, 2007. 3, 4, 5, 44

[FJ15] John D. Faben and Mark Jerrum. The Complexity of Parity Graph Homomorphism: An Initial Investigation. *Theory Comput.*, 11:35–57, 2015. 43

[FMR24] Jacob Focke, Dániel Marx, and Paweł Rzążewski. Counting List Homomorphisms from Graphs of Bounded Treewidth: Tight Complexity Bounds. *ACM Trans. Algorithms*, 20:11:1–11:32, 2024. 43

[FV98] Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.*, 28(1):57–104, 1998. 43

[GCD23] Artem Govorov, Jin-Yi Cai, and Martin E. Dyer. A Dichotomy for Bounded Degree Graph Homomorphisms with Nonnegative Weights. *J. Comput. Syst. Sci.*, 132:1–15, 2023. 43

[GGJT10] Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A Complexity Dichotomy for Partition Functions with Mixed Signs. *SIAM J. Comput.*, 39:3336–3402, 2010. 43

[GGR14] Andreas Göbel, Leslie Ann Goldberg, and David Richerby. The Complexity of Counting Homomorphisms to Cactus Graphs Modulo 2. *ACM Trans. Comput. Theory*, 6:17:1–17:29, 2014. 43

[GGR16] Andreas Göbel, Leslie Ann Goldberg, and David Richerby. Counting Homomorphisms to Square-Free Graphs, Modulo 2. *ACM Trans. Comput. Theory*, 8:12:1–12:29, 2016. 43

[GHK+24] Robert Ganian, Thekla Hamm, Viktoriia Korchemna, Karolina Okrasa, and Kirill Simonov. The Fine-Grained Complexity of Graph Homomorphism Parameterized by Clique-Width. *ACM Trans. Algorithms*, 20:19:1–19:26, 2024. 43

[GLSY23] Lior Gishboliner, Yevgeny Levanzov, Asaf Shapira, and Raphael Yuster. Counting Homomorphic Cycles in Degenerate Graphs. *ACM Trans. Algorithms*, 19(1):2:1–2:22, 2023. 4

[GMN+24] Carla Groenland, Isja Mannens, Jesper Nederlof, Marta Piecyk, and Paweł Rzążewski. Towards Tight Bounds for the Graph Homomorphism Problem Parameterized by Cutwidth via Asymptotic Matrix Parameters. In *ICALP*, pages 77:1–77:21, 2024. 43

[Grö59] H Grötzsch. Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel. *Wiss. Z. Martin Luther Univ. Halle-Wittenberg, Math. Nat. Reihe*, 8:109–120, 1959. 18

[Gro07] Martin Grohe. The Complexity of Homomorphism and Constraint Satisfaction Problems Seen from the Other Side. *J. ACM*, 54:1, 2007. 42

[GT11] Martin Grohe and Marc Thurley. Counting Homomorphisms and Partition Functions. In *Model Theoretic Methods in Finite Combinatorics*, pages 243–292. American Mathematical Society, 2011. 43

[Har64] Lawrence Hueston Harper. Optimal assignments of numbers to vertices. *Journal of the Society for Industrial and Applied Mathematics*, 12(1):131–135, 1964. 40

[HN90] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Theory B*, 48(1):92–110, 1990. 43

[HN92] Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discret. Math.*, 109(1-3):117–126, 1992. 43

[HN26] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms.* Oxford University Press, 2026. 3, 44

[HS70] András Hajnal and Endre Szemerédi. Proof of a Conjecture of Erdős. In *Combinatorial Theory and Its Applications II*, pages 601–623. North-Holland, 1970. 10

[HT10] Thore Husfeldt and Nina Taslaman. The Exponential Time Complexity of Computing the Probability That a Graph Is Connected. In *IPEC*, volume 6478 of *Lecture Notes in Computer Science*, pages 192–203. Springer, 2010. 44

[HW12] Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *SODA*, pages 104–113. SIAM, 2012. 4

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62:367–375, 2001. 8

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63:512–530, 2001. 5, 8, 28, 33

[JVW90] François Jaeger, Dirk L Vertigan, and Dominic JA Welsh. On the computational complexity of the Jones and Tutte polynomials. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 108, pages 35–53. Cambridge University Press, 1990. 44

[KKMS10] Henry A. Kierstead, Alexandr V. Kostochka, Marcelo Mydlarz, and Endre Szemerédi. A fast algorithm for equitable coloring. *Comb.*, 30(2):217–224, 2010. 10

[KKP25] Balagopal Komarath, Anant Kumar, and Akash Pareek. Counting Patterns in Degenerate Graphs in Constant Space. *CoRR*, abs/2511.04258, 2025. 4

[KLR03] Martin Kochol, Vadim V. Lozin, and Bert Randerath. The 3-Colorability Problem on Graphs with Maximum Degree Four. *SIAM J. Comput.*, 32(5):1128–1139, 2003. 8

[KM24] Alexander S. Kulikov and Ivan Mihajlin. If Edge Coloring Is Hard Under SETH, Then SETH Is False. In *Symposium on Simplicity in Algorithms*, pages 115–120, 2024. 5, 29, 36, 46

[KMPS24] Karthik C. S., Dániel Marx, Marcin Pilipczuk, and Uéverton S. Souza. Conditional Lower Bounds for Sparse Parameterized 2-CSP: A Streamlined Proof. In *Symposium on Simplicity in Algorithms*, pages 383–395, 2024. 5, 6, 23

[Lar02] Benoît Larose. Families of Strongly Projective Graphs. *Discuss. Math. Graph Theory*, 22:271–292, 2002. 43

[Law76] Eugene L. Lawler. A Note on the Complexity of the Chromatic Number Problem. *Inf. Process. Lett.*, 5(3):66–67, 1976. 3

[ŁN04] Tomasz Łuczak and Jaroslav Nešetřil. Note on Projective Graphs. *J. Graph Theory*, 47:81–86, 2004. 43

[Lov12] László Lovász. *Large Networks and Graph Limits.* American Mathematical Society, 2012. 3

[LT01] Benoît Larose and Claude Tardif. Strongly Rigid Graphs and Projectivity. *Mult.-Valued Log.*, 7:339–361, 2001. 43

[LY22] Jiatu Li and Tianqi Yang. $3.1n - o(n)$ Circuit Lower Bounds for Explicit Functions. In *STOC*, pages 1180–1193, 2022. 8

[Mar10] Dániel Marx. Can You Beat Treewidth? *Theory Comput.*, 6:85–112, 2010. 5, 23, 43

[MB83] David W. Matula and Leland L. Beck. Smallest-Last Ordering and clustering and Graph Coloring Algorithms. *J. ACM*, 30(3):417–427, 1983. 4

[MO25] Sebastian Meyer and Jakub Opršal. A Topological Proof of the Hell-Nešetřil Dichotomy. In *SODA*, pages 4507–4519, 2025. 43

[OPR20] Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full Complexity Classification of the List Homomorphism Problem for Bounded-Treewidth Graphs. In *European Symposium on Algorithms*, pages 74:1–74:24, 2020. 43

[OR21] Karolina Okrasa and Paweł Rzążewski. Fine-Grained Complexity of the Graph Homomorphism Problem for Bounded-Treewidth Graphs. *SIAM J. Comput.*, 50:487–508, 2021. 43

[PRS09] Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Solving Dominating Set in Larger Classes of Graphs: FPT Algorithms and Polynomial Kernels. In *ESA*, Lecture Notes in Computer Science, pages 694–705. Springer, 2009. 4

[PRS12] Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Trans. Algorithms*, 9(1):11:1–11:23, 2012. 4

[PS24] Daniel Paul-Pena and C. Seshadhri. A Dichotomy Theorem for Linear Time Homomorphism Orbit Counting in Bounded Degeneracy Graphs. In *ISAAC*, volume 322 of *LIPIcs*, pages 54:1–54:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. 4

[PS25] Daniel Paul-Pena and C. Seshadhri. Subgraph Counting in Subquadratic Time for Bounded Degeneracy Graphs. In *ICALP*, volume 334 of *LIPIcs*, pages 124:1–124:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. 4

[PS26] Daniel Paul-Pena and C. Seshadhri. Near-Linear Time Subhypergraph Counting in Bounded Degeneracy Hypergraphs. In *SODA*, pages 3652–3687, 2026. 4

[RS86] Neil Robertson and Paul D. Seymour. Graph Minors. II. Algorithmic Aspects of Tree-Width. *J. Algorithms*, 7(3):309–322, 1986. 40

[Rzą14] Paweł Rzążewski. Exact Algorithm for Graph Homomorphism and Locally Injective Graph Homomorphism. *Inf. Process. Lett.*, 114:387–391, 2014. 3, 4, 44

[Sha49] Claude E. Shannon. The Synthesis of Two-Terminal Switching Circuits. *Bell Syst. Tech. J.*, 28(1):59–98, 1949. 7

[Tra08] Patrick Traxler. The Time Complexity of Constraint Satisfaction. In *Parameterized and Exact Computation*, pages 190–201, 2008. 3, 5

[Tut54] William Thomas Tutte. A contribution to the theory of chromatic polynomials. *Canadian journal of mathematics*, 6:80–91, 1954. 44

[Wah11] Magnus Wahlström. New Plain-Exponential Time Classes for Graph Homomorphism. *Theory Comput. Syst.*, 49:273–282, 2011. 3, 4, 44

[Wil18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de Janeiro 2018*, pages 3447–3487. World Scientific, 2018. 8

[Zhu17] Dmitriy Zhuk. A Proof of CSP Dichotomy Conjecture. In *FOCS*, pages 331–342. IEEE Computer Society, 2017. 43

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il