



Graph Isomorphism and Representation Theory

Joshua A. Grochow and Jacob Urisman

June 2026

Abstract

We introduce an approach to distinguishing isomorphism types of graphs based on vector spaces of polynomials that are set-wise invariant under permutations (so-called separating modules, which are, in particular, representations of the symmetric group), inspired by the Geometric Complexity Theory approach to separating complexity classes (Mulmuley & Sohoni, *SIAM J. Comput.*, 2001). We characterize the power of this method for distinguishing non-isomorphic graphs under several different complexity measures:

- Degree. We show that separating modules of “support-degree” k (each monomial touches at most k vertices) are equivalent in power to the counts of $O(k)$ -vertex subgraphs. This is strictly weaker than $O(k)$ -dimensional Weisfeiler–Leman by a previous result of Fürer (ICALP ’01).
- Symmetric algebraic circuit size. If we measure the complexity of a separating module by the size of the smallest multi-output symmetric algebraic circuit that computes a spanning set for it, we show that separating modules of symmetric circuit size $n^{\Theta(k)}$ are equivalent to $\Theta(k)$ -WL. This generalizes and strengthens the result of Dawar & Wilsenach (CSL ’18; ICALP ’20; *ACM Trans. Comput. Log.*, 2022; *Theory Comput.*, 2025): they proved one direction of this equivalence, and only for invariant polynomials, whereas we generalize to separating modules and prove both directions.
- Multiplicities. When considering only the representation-theoretic multiplicities of separating modules (as was proposed in GCT by Mulmuley & Sohoni, *ibid.*, rather than the polynomials themselves), we show that two graphs are separated by multiplicities if and only if their automorphism groups have different multiplicity of cycle types (cycle index).

The latter result is notable in the analogy with GCT, as it is the only result we are aware of in which the multiplicity approach to separating isomorphism types of objects has been given an exact “intrinsic” characterization in terms of the objects themselves. We use this characterization to show that for graphs, multiplicity obstructions are stronger than occurrence obstructions. We also make connections between invariant polynomials, the Graph Reconstruction Conjectures (Kelly, *Pacific J. Math.*, 1957; Ulam, 1960; Harary, 1964), and Forman’s “invariants of finite type” (*Adv. Math.* 2004).

Contents

1	Introduction	2
2	Preliminaries	10
3	Degree of separating modules, invariants, and subgraph census	14
4	Support size and complexity measures on the symmetric group	20
5	Symmetric circuit size of separating modules and Weisfeiler–Leman	22

6	Multiplicity obstructions for graph non-isomorphism	27
7	Separating invariants and the Reconstruction Conjectures	36
A	Code for the calculation in Proposition 6.11	39

1 Introduction

The GRAPH ISOMORPHISM (GI) problem is one of the most long-standing problems in complexity theory, appearing in both Cook’s and Levin’s original papers on NP [Coo71, Lev73] (see [AD17a] for the history). It is one of only a handful of problems in the landscape of complexity theory that has no known polynomial-time algorithm and is also not known to be NP-complete. Graph isomorphism has many practical applications as graphs are an extremely useful data structure, especially for network modeling. The problem is as follows: given two graphs G, H , decide whether there some relabeling of nodes of G which makes it equal to H . More formally two graphs G, H are isomorphic, denoted $G \cong H$, if there exists a bijection $f : V(G) \rightarrow V(H)$ such that $(u, v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(H)$, and the algorithmic problem is to detect, given G, H , whether an isomorphism exists.

A common way to distinguish non-isomorphic graphs is to find some (*isomorphism*) *invariant* of the graphs that differs between them: a function f on labeled graphs (or adjacency matrices) such that $f(G) = f(H)$ whenever $G \cong H$. If $f(G) \neq f(H)$, then certainly G and H are not isomorphic, but it is possible for some invariant f that $G \not\cong H$ yet $f(G) = f(H)$. Easy-to-compute invariants include the multiset of vertex degrees, or whether the graph is connected. A harder-to-compute invariant would be the number of 3-colorings. Generally we seek invariants that are easy to compute.

An analogous situation arises in complexity lower bounds: to show that some hard function f_{hard} is not in some complexity class \mathcal{C}_{easy} , one often seeks a property of functions that is invariant under some natural group of symmetries (such as permuting the inputs to the functions, which often doesn’t change their complexity, or sometimes even taking linear combinations of inputs), show that every function in \mathcal{C}_{easy} has the property, but f_{hard} does not. In the analogy, we may think of f_{hard} and each $f_{easy} \in \mathcal{C}_{easy}$ as analogous to two graphs we want to distinguish. When \mathcal{C}_{easy} has a complete problem the analogy is even tighter: if f_{easy} is \mathcal{C}_{easy} -complete, then one is typically seeking to show that f_{hard} is not in the topological closure of the GL_n -orbit of f_{easy} . In the GI case, one can also phrase the problem as showing that G is not in the topological closure of the S_n -orbit of H , but because the orbits (=isomorphism classes) are finite, the orbit itself is already topologically closed. This is one of the ways in which GI is “simpler” in this analogy. See Lysikov & Walter [LW25] for a recent paper developing the complexity theory of orbit and orbit closure problems.

The Geometric Complexity Theory (GCT) program, initiated by Mulmuley & Sohoni [MS01], takes this idea, and proposes specific ways to use algebraic geometry (seeking invariant properties defined by the vanishing of polynomials) and representation theory (the way the relevant symmetry group acts on the ring of polynomials) to help find such invariant properties.

In this paper, we use this analogy both as an opportunity to explore potentially new methods for separating non-isomorphic graphs, and as an opportunity to use the “simpler” setting of GI (separating pairs of non-isomorphic graphs seems easier than separating complexity classes) as a testbed for ideas and techniques that first arose in GCT.

Although the methods we develop do not end up being stronger than previously-proposed methods for GI (specifically, the Weisfeiler–Leman family of algorithms, about which we say more below), our exploration nonetheless reveals a number of interesting new connections between GI, algebraic proof complexity, the Graph Reconstruction Conjectures, and the representation-theoretic approach to separations put forward in GCT. In the remainder of this introduction we motivate the approach, explain the mathematical objects involved (primarily representations of the symmetric group), and put our main results in context.

In order to apply algebraic geometry in the setting of GI, we deal with polynomial equations. Although prior work by many authors [BG15, BG17, Der13, SSC14, AM13] has also dealt with a system of polynomial equations, we take a different approach. In prior works (*ibid.*), they typically have variables x_{ij} , and polynomial equations in those variables whose vanishing enforces that the $n \times n$ matrix X is a permutation matrix that gives an isomorphism between G and H . If G and H are not isomorphic, then the resulting system of polynomial equations is not satisfiable, and one can use various algebraic or semi-algebraic proof systems to try to refute the satisfiability of the system of equations—equivalently, refute the existence of an isomorphism between G and H . Several complexity measures on those algebraic proof systems turn out to be equivalent to or stronger than WL (*ibid.*).

In contrast, in our setting, we wish to examine how to use polynomials “on the graphs themselves” to distinguish non-isomorphic graphs. That is, we also have variables x_{ij} , but the variable x_{ij} for us just gives the (i, j) entry of the adjacency matrix of G , rather than encoding part of a potential isomorphism between G and H . We can then ask what isomorphism-invariant properties of graphs can be expressed this way. As a simple example, $\sum_{i,j} x_{ij}$ is clearly invariant, and computes the total number of edges of a graph. (Since we ultimately will wish to use the vanishing or non-vanishing of polynomials to distinguish non-isomorphic graphs, we note that the invariant $-|E(G)| + \sum_{i,j} x_{ij}$ vanishes on G , but not on any graph with a different number of edges.)

However, there are some invariant properties of graphs that are more naturally described by the vanishing of *sets* of polynomials rather than individual invariant polynomials. For example, the invariant polynomial

$$\sum_{i,i' \in [n], i \neq i'} \left(\sum_{j \neq i} x_{ij} - \sum_{j \neq i'} x_{i'j} \right)^2$$

vanishes if and only if all vertices have the same out-degree (we assume throughout the paper that we work over a field of characteristic zero). But in fact how this polynomial is built is as a sum of squares, and since adjacency matrices are $\{0, 1\}$ -valued, all terms evaluate to real values, so a sum of squares vanishes iff the individual summands vanish. It is arguably more natural to say that a graph is out-regular if every polynomial in the set

$$\left\{ \sum_{j \neq i} x_{ij} - \sum_{j \neq i'} x_{i'j} : i, i' \in [n], i \neq i' \right\}$$

vanishes on its adjacency matrix. (Note that this set of polynomials also has lower degree than the single polynomial above; degree is one of the relevant notions of complexity in our approach, as we will see.)

What is it that makes a given set of polynomials define an isomorphism-invariant property? To see this, we can use the standard rephrasing of GI as an “orbit problem:” the symmetric group S_n —consisting of permutations of $[n]$ —acts on $n \times n$ adjacency matrices by permuting the indices: $(\pi \cdot A)_{ij} = A_{\pi^{-1}(i), \pi^{-1}(j)}$, and then two adjacency matrices A, B correspond to isomorphic graphs iff there exists $\pi \in S_n$ such that $\pi \cdot A = B$. (If P_π is the corresponding permutation matrix, then the action of π on A is given by $\pi \cdot A = P_\pi A P_\pi^\top$.) There is then a “dual action” of S_n on the variables x_{ij} : $\pi \cdot x_{ij} = x_{\pi^{-1}(i), \pi^{-1}(j)}$, which has the effect of, for any polynomial f , $(\pi \cdot f)(A) = f(\pi^{-1} \cdot A)$. If S is a set of polynomials, then the vanishing of all polynomials in S defines an invariant property of graphs if the action of S_n permutes the set S amongst itself: for if $\{\pi \cdot f : f \in S\} = S$, and $G \cong H$, then there is some π such that $\pi \cdot G = H$, and we have $f(H) = f(\pi \cdot G) = (\pi^{-1} \cdot f)(G)$, which is some other function in S evaluated at G . So all the functions in S vanish on G iff each function in S vanishes on all graphs isomorphic to G . (An invariant polynomial as above is then just a singleton set S .)

By a similar proof, even if $\pi \cdot S$ is just contained in the linear span of the polynomials in S , the vanishing of S will still define an invariant property. This motivates a key definition. Since we

will only work with $\{0, 1\}$ -adjacency matrices, we need only consider multilinear polynomials (every variable appears in each monomial with degree ≤ 1), or equivalently, work in the quotient of the polynomial ring $\mathbb{F}[x_{ij} : i, j \in [n]]/\langle x_{ij}^2 - x_{ij} : i, j \in [n] \rangle$. The idea of this definition in the context of complexity classes goes back to [MS01]; we follow the nomenclature and definition from [Gro15].

Definition 1.1 (Test module, separating module, cf. [Gro15, Def. 2.7]). A *test module* (for graph isomorphism) is a vector space $T \subseteq \mathbb{F}[x_{ij} : i, j \in [n]]/\langle x_{ij}^2 - x_{ij} : i, j \in [n] \rangle$ that is closed under the above action of S_n .¹ for all $f \in T$ and $\pi \in S_n$, we have $\pi \cdot f \in T$ as well.

Given two graphs G, H , a test module T is called a *separating module* for G and H , or is said to *separate G from H* , if all $f \in T$ vanish on A_G , and some $f \in T$ does not vanish on A_H .

As we have argued above, the set of graphs on which a given test module T vanishes is an isomorphism-invariant set of graphs. Conversely, the set of all polynomials that vanish on the set of all graphs isomorphic to G is a test module. These will be our main object of exploration in this paper. We now come to our results.

Degree. We were motivated to explore the distinguishing power of separating modules using degree as a complexity measure by the work of Berkholz and Grohe. In [BG15], they showed an equivalence between degree in the proof system they dubbed monomial-PC and WL dimension.

To state the simplest version of our first result, we introduce the notion of “support-degree” of a monomial in the variables x_{ij} : the support-degree of a monomial is the number of distinct vertices referenced in its subscripts, and the support-degree of a polynomial is the maximum support-degree of any monomial in it. By the “census” of d -vertex subgraphs, we mean the multiset of isomorphism types of d -vertex subgraphs.

Theorem 1.2 (Simplified version of Thm. 3.11). *Two graphs are distinguished by separating modules of support-degree d if and only if they are distinguished by separating invariants of support-degree $\Theta(d)$ if and only if they are distinguished by the census of $\Theta(d)$ -vertex subgraphs.*

In particular, it follows immediately from a lower bound of Fürer [Für01] that the proof systems mentioned above are strictly weaker than $\Theta(d)$ -dimensional Weisfeiler–Leman, as the subgraph census is the initial coloring of WL, and Fürer shows that a linear number of rounds of refinement are required in the worst case.

We also connect invariant degree to the (in)famous Reconstruction Conjectures [Kel42, Kel57, Ula60, Har64]. A graph is called k -reconstructible (resp., edge-reconstructible) if its isomorphism type is uniquely identified by the multiset of isomorphism types of its k -vertex-deleted subgraphs (resp., k -edge-deleted). The Reconstruction Conjecture is that all n -vertex graphs are $(n - 1)$ -reconstructible, which, remarkably, remains unproved. In Theorems 7.6 and 7.5 we show that a graph is $(n - k)$ -reconstructible if and only if it is identified by invariants of support-degree k (and analogously for edge-reconstructible and degree). These results were previously established implicitly, in different language, by Forman [For04]; see Section 1.1 below for a more detailed comparison.

What is it that leads to the difference between monomial-PC degree [BG15] and degree of separating modules, in terms of their power to distinguish graphs? As discussed above, in the setting of Berkholz & Grohe, the meaning of their variable x_{ij} is that vertex i in graph G maps to vertex j in graph H . Therefore, a degree k polynomial allows them to reason about possible partial isomorphisms from k vertices of G to k vertices of H . In contrast, in our setting, the variables are x_{ij} which takes value 1 if there is an edge from node i to node j in graph G . Therefore degree k polynomials in our setting only let us reason about k -edge subgraphs of G , and separately k -edge subgraphs of H , but not directly the relations between them. We will see next that when we measure complexity by symmetric algebraic circuit size instead of degree, separating modules do become powerful enough to simulate WL.

¹We will have occasion to consider test modules contained in the un-quotiented polynomial ring $\mathbb{F}[x_{ij} : i, j \in [n]]$ as well, and for those the definition is the same, *mutatis mutandis*.

Symmetric circuit size. As context for our next result, we first recall the Weisfeiler–Leman family of algorithms. The Weisfeiler–Leman (WL) algorithm for graph isomorphism was first formulated in 1968 by Weisfeiler and Leman [WL68]. The k -dimensional WL algorithm works by coloring k -tuples of vertices according to the isomorphism type of their induced subgraph, and then iteratively refining the coloring based on the colors of “neighboring” k -tuples; this algorithm can be implemented in $n^{O(k)}$ time. However, Cai, Fürer, and Immerman [CFI92] showed that there are n -vertex graphs that can only be separated by $\Omega(n)$ -dimensional WL (which would take $n^{\Omega(n)}$ time, more than the $n!$ of the trivial algorithm). Nonetheless, WL has been a useful subroutine in many other algorithms for GI, notably in Babai’s quasi-polynomial-time algorithm [Bab16]. WL has many characterizations in terms of coloring, first-order logical properties of graphs [CFI92], and proof complexity [AM13, BG15, BG17, TW23].

Motivated by the results of Dawar and Wilsenach, we then look at symmetric circuit size as a complexity measure on separating modules. In [DW25], they show that WL can simulate the computation of symmetric algebraic circuits. To relate separating modules to symmetric algebraic circuits, we define multi-output circuits which compute a spanning set of polynomials of a separating module. We then show that k -WL is equivalent in distinguishing power to separating modules of symmetric circuit size $n^{\Theta(k)}$.

Theorem 1.3 (Simplified version of Cor. 5.11). *Two graphs are distinguished by separating modules (resp., separating invariants) computed by symmetric algebraic circuits of size $n^{\Theta(k)}$ if and only if they are distinguished by $\Theta(k)$ -WL.²*

Given that WL is essentially a proof system for graph non-isomorphism, that is equivalent to various algebraic proof systems such as monomial-PC [BG15], it is natural to wonder about algebraic proof systems based on symmetric algebraic circuits. This was studied recently by Dawar, Grädel, Kullmann, and Pago [DGKP25]. Similar to previous works, their starting point is the system of equations encoding all possible isomorphisms, rather than in our setting where our polynomials encode properties of individual (isomorphism classes of) graphs.

As a side benefit, we also relate the notion of symmetric algebraic circuit size of a test module to other notions of complexity on the symmetric group, such as sensitivity, block-sensitivity, and decision tree complexity [DFL⁺21]; see Prop. 4.4.

Multiplicity obstructions. In the first paper on GCT [MS01], they propose separating complexity classes by using only multiplicities instead of separating modules or polynomials. To explain this approach in our setting, we consider two separating modules to be “abstractly isomorphic” if S_n “acts on them in the same way,” regardless of which actual polynomials they contain. More specifically, two test modules T, T' are said to be *equivalent* if there is a bijective linear map $f: T \rightarrow T'$ such that $f(\pi u) = \pi f(u)$ for all $u \in T, \pi \in S_n$. For any given test module T , one can then ask how many test modules *equivalent to T* vanish on a given graph G , and this is called the multiplicity of T in the vanishing ideal of G ’s orbit. If there is some abstract equivalence class of test modules that has a different multiplicity for G than for H , then certainly they are not in the same orbit, hence are not isomorphic. Such an equivalence class of test modules is called a *multiplicity obstruction*, as its existence obstructs the existence of an isomorphism. The multiplicity approach to separating graphs thus asks whether the number of vanishing separating modules of each type is equivalent between two graphs.

In this setting, we get a clean intrinsic/combinatorial characterization of what these representation-theoretic multiplicities tell us about graph isomorphism. This is the first characterization of this kind we are aware of in a GCT-style multiplicity approach to separating orbit(closure)s.

²In more detail, this equivalence holds for all constant k ; as k approaches n^c the factors in the big-Theta depend on c in a way we make explicit in Cor. 5.11. For example, for k up to $\sqrt{n}/2$, we still get an equivalence with a factor of 2 in the big-Theta.

Theorem 1.4 (=Cor. 6.6). *Two n -vertex graphs G and H admit a multiplicity obstruction if and only if the counts of cycle types of elements in $\text{Aut}(G) \leq S_n$ and $\text{Aut}(H) \leq S_n$ are distinct.*

In particular, since most graphs have trivial automorphism group, most graphs are not distinguished by multiplicity obstructions (Obs. 6.7). More generally, if two n -vertex graphs have automorphism groups that are conjugate in S_n , then they are indistinguishable by multiplicity obstructions. It is an interesting open question whether this is the only obstacle to being distinguished by multiplicity obstructions. We say more about this question in Section 1.2 below.

In the absence of an answer to the preceding question, we find it natural to ask whether there are interesting families of graphs—with interesting automorphism groups—that are distinguished by multiplicity obstructions. One such possible family of graphs is suggested again by analogy with GCT. Part of the philosophy behind GCT is that the representation-theoretic, and especially multiplicity-based, approach is only expected to work for objects (polynomials, tensors, graphs, etc.) that are *characterized by their symmetries*, a notion introduced in [MS01, Sec. 4]. If V is a representation of a group G , a (nonzero) point $v \in V$ is said to be *characterized by its symmetries* if for all points $u \in V$, if $\text{stab}_G(v) \subseteq \text{stab}_G(u)$ —that is, for all $g \in G$, if $gv = v$ then $gu = u$ —then u must be a scalar multiple of v . In GCT, notable examples of symmetry-characterized polynomials include the determinant and permanent [MS01, Props. 4.2 and 4.5], and since then many other polynomials of complexity-theoretic interest have been shown to be characterized by their symmetries (e. g., [CKW10, Gro12b, Gro12a, RRR19]).

But a graph and its complement have the same automorphism group, so for a graph G to be “characterized by its symmetries” we mean that for any graph H on the same number of vertices, if $\text{Aut}(G) \leq \text{Aut}(H)$ (as subgroups of S_n), then H must be isomorphic to either G or its complement (or the empty or complete graphs, since every permutation group is a subgroup of their automorphism groups; these play a role much like the zero vector in a vector space).³ It is an interesting question whether multiplicity information alone can distinguish all pairs of non-isomorphic symmetry-characterized graphs. In Prop. 6.8 we make a small inroad into this question: using the theory of representation stability and FI-modules [CF13, CEF15], we show that *constant-degree* multiplicity obstructions can distinguish at most constantly many isomorphism types. The more general question of unbounded-degree multiplicity obstructions remains open; see Section 1.2 for more on this.

Occurrences obstructions. In [MS01], one of the original conjectures (since disproven in the context of the Permanent vs Determinant Conjecture by Bürgisser, Ikenmeyer, and Panova [BIP19]), was to not only use multiplicity obstructions to separate complexity classes, but actually to use so-called “occurrence” obstructions, which are multiplicity obstructions in which the multiplicity is *zero* for one class and *non-zero* for the other. In the context of GCT, Dörfler, Ikenmeyer and Panova [DIP20] showed that there are settings in which multiplicity obstructions are strictly more powerful than occurrence obstructions for separations. We show that the same is true for graphs:

Proposition 1.5 (Simplified version of Proposition 6.11). *There are infinitely many pairs of graphs that are distinguished by multiplicity obstructions but not by occurrence obstructions.*

1.1 Related work

As far as we are aware, ours is the first study using separating modules for graph isomorphism. The only other paper we are aware of that even uses representation theory in the context of GI is Derksen

³Let V be the vector space of adjacency matrices, and J be the adjacency matrix of the complete graph, and $U := V/J$ the quotient vector space. Then in U , a graph and its complement are scalar multiples of one another: $A_{G^c} = J - A_G \equiv -A_G \pmod{J}$. And among simple graphs, a graph is only a nonzero scalar multiple of itself and its complement. Furthermore, the empty graph and the complete graph both correspond to the zero vector in U . So in the space U , we can use the now-standard definition of symmetry-characterization mentioned above, and see that it agrees with the natural notion that one might want for graphs.

[Der13], which has quite a different flavor to what we pursue here. Indeed, as with the other prior work, its starting point is a system of equations whose solutions encode all possible isomorphisms, rather than looking at (sets of) polynomials on a single graph at a time.

Related work on invariants. We are aware of three lines of research using what we would call separating (polynomial) *invariants* for graph isomorphism. The first of these three goes back implicitly to Kocay [Koc82], who considered an “algebra of subgraphs” based on the counts of induced or general subgraphs. However, Kocay seems not to have formulated his work in terms of invariant polynomials, and the identities between such numerical invariants he considers are all linear.

The second, and closer to our work, is that of Pouzet [Pou77, Pou79], continued more recently with Thiéry [Thi00, PT01]. In their work they consider polynomial invariants of *weighted* adjacency matrices, that is, arbitrary matrices over a field of characteristic zero (rather than just $\{0, 1\}$ matrices or unweighted graphs as we consider). Effectively, they are working in the (infinite-dimensional) ring $\mathbb{F}[x_{ij}]$ whereas we (mostly) work in the (finite-dimensional) ring $\mathbb{F}[x_{ij}]/\langle x_{ij}^2 - x_{ij} \rangle$. The invariant ring in that setting is, expectedly, more complicated, having a generating set in terms of counting multi-graphs (with non-negative integer weights on the edges) rather than in terms of counting ordinary graphs (as in our setting). The focus on weighted graphs makes their work connected to weighted/algebraic generalizations of the Reconstruction Conjectures, but not (as far as we are aware) to the original (unweighted) Reconstruction Conjectures.

The third work on invariants, and closest to our work, is that of Forman. Forman [For04] introduced what he called “graph invariants of (edge- or vertex-)finite type”, which had an “order” k , and used these to prove a number of results on the Graph Reconstruction Conjectures. Forman’s “invariants of finite type” are isomorphism-invariant functions on the (infinite) set of all isomorphism types of finite simple graphs satisfying certain inclusion-exclusion-like identities. By our Observation 3.10 and Forman’s Theorem 3.3, Forman’s invariants of edge-finite-type and order k , applied to n -vertex graphs, are precisely given by invariants (in the usual polynomial sense studied in our paper) of degree k . Similarly, by Observation 3.10 and Forman’s Theorem 10.2, Forman’s invariants of vertex-finite-type and order k , applied to n -vertex graphs, are precisely given by invariants in our sense of what we call “support-degree” k (which counts the number of vertices involved in a monomial rather than the number of variables=edges). Forman’s Theorems 4.6 and 11.4 are essentially equivalent, via the preceding correspondence, to our connections between invariant polynomials and the Reconstruction Conjectures. Once we get to the point of Observation 3.10 and Forman’s Theorems 3.3 and 10.2, our proofs of the connections with the Reconstruction Conjectures are essentially the same as his, though we came to them independently.

We also remark that some results in Forman’s paper may have shorter proofs in the language of invariant polynomials (indeed, Observation 3.10 is a pretty trivial observation, whereas Forman’s Theorems 3.3 and 10.2 take some work), while the opposite may be true for other results.

Technically, Forman also deals with invariants on the infinite set of all graphs—similar to Kocay (*ibid.*)—whereas we focus on invariant polynomials on graphs with n vertices (asymptotically as $n \rightarrow \infty$, but for each n we have a finitely generated algebra). Translating Forman’s results more faithfully into the language of polynomials seems to require something like invariant polynomials on graphs with infinitely many vertices (analogous to the ring of symmetric functions on infinitely many variables) or FI-rings [CF13, CEF15]; we hope to pursue such development in future work.

Related work on Weisfeiler–Leman, symmetric algebraic circuits, and proof complexity. Finally, we discuss several works related to WL and proof complexity that are somewhat related but quite distinct from our work.

Berkholz and Grohe [BG15] introduce an algebraic proof system they name monomial-PC, and show that its distinguishing power is equivalent to WL. In particular, they start by setting up a system of polynomials which encode all possible isomorphisms between two graphs and set those

polynomials to zero. They then use polynomial calculus proof system on that system of polynomials while restricting certain intermediate steps. They show that if a refutation exists, then one exists where the maximum degree of any equation derived in that refutation is equal to the WL dimension required to distinguish the two graphs. There are several other algebraic and semi-algebraic proof systems that are equivalent to WL when applied to this same system of polynomials (or closely related systems of inequations) [SSC14, AM13, BG17].

Toran and Wörz [TW23] work in the setting of Boolean proof complexity. They use the Boolean version of essentially the same system of equations as in [BG15], and apply the Narrow Resolution proof system to refute the system of equations. They show that this proof system, under the complexity measure of narrow resolution depth and width, is equivalent in distinguishing power to WL. It is possible that there is a direct connection between symmetric circuit size and support size in our setting and narrow resolution depth and width. Indeed, one can develop a Boolean analogue of separating modules in the setting of CNFs, consisting of a set of Boolean formulas that is set-wise invariant under the action of S_n . It would be potentially interesting to explore this potential connection, but we do not do so in our paper.

Anderson & Dawar [AD17b] show that any first-order structure on which properties can be computed by Boolean (threshold) circuits can be simulated by first-order formulas. Their proof preserves circuit size as number of variables and circuit depth as number of quantifiers. This allowed us to show equivalence between symmetric algebraic circuits and the first order formulas for graph isomorphism.

Dawar & Wilsenach [DW25] show that WL can be simulated by symmetric algebraic circuits. In their paper, they rely on multiple background results proved thoroughly in [DW22] and [AD17b]. The former of those contains results for rank logic which we do not explore in this paper. We prove the converse of their results for symmetric algebraic circuits.

Theorem 1.3 (=Corollary 5.11) is connected to several other results in the literature on subgraph counts. Say that a function f on isomorphism types of graphs is k -WL invariant if whenever G and G' are k -WL indistinguishable, $f(G) = f(G')$. Nueun [Neu24] proves that the counts of H as a subgraph is k -WL-invariant if and only if $k \geq hdtw(H)$, where $hdtw$ denotes the hereditary treewidth (we refer to *ibid.* for definitions). This count is captured by an invariant polynomial we denote S_H (see Section 3.3). Consistent with Corollary 5.11 and Neuen’s aforementioned result, Dawar, Pago, and Seppelt [DPS26] proved that the invariant polynomial S_H can be computed by symmetric algebraic circuits of size $n^{O(hdtw(H))}$ (combine Theorems 1.1 and 7.1 of the arXiv version of their paper).⁴

1.2 Future directions and open questions

The connections we develop in this paper also raise a number of intriguing new questions, some of which we raise here.

We have already mentioned this question above, but we highlight it here and discuss it a little further:

Open Question 1.6. Is it the case that for all pairs of n -vertex graphs (G, H) such that $\text{Aut}(G)$ and $\text{Aut}(H)$ are not conjugate in S_n , G and H are distinguished by multiplicity obstructions?⁵

There exist pairs of subgroups of S_n that are not conjugate but have the same cycle index;⁶

⁴Technically their results are for bipartite graphs and bipartition-respecting homomorphisms/subgraphs, but they directly transfer to the non-bipartite case.

⁵This question is interesting in any of the usual graph variants: looped or loopless, directed or undirected, bipartite (with the action of $S_n \times S_m$), etc. In the case of multi-graphs—where each edge can occur with multiplicity ≥ 1 —one should consider multiplicities in the un-quotiented polynomial ring $\mathbb{F}[x_{ij} : i, j \in [n]]$.

⁶By direct calculation, we found that the smallest n with such an example in S_n is $n = 8$, where $\langle (1, 8, 5, 4)(2, 3, 7, 6), (2, 7)(3, 6), (1, 6)(2, 8)(3, 5)(4, 7) \rangle \cong C_4 \circ D_4$, where \circ denotes a central product, and $\langle (2, 7)(3, 6), (1, 7, 4, 6)(2, 8, 3, 5) \rangle \cong C_2^2 \rtimes C_4$ are not even abstractly isomorphic—hence, *a fortiori*, not conjugate—but have the same cycle indices.

however, we do not know if there exist such examples that are also the automorphism groups of (simple, uncolored) graphs. If not, that would imply a positive answer to Question 1.6.

Next we highlight a related question from above:

Open Question 1.7. Is every pair of graphs that are characterized by their symmetries distinguished by multiplicity obstructions? Is every graph characterized by its symmetries also characterized by its multiplicities?

If a symmetry-characterized graph is not trivial (complete or empty graph) nor a star graph (complete bipartite graph $K_{n,1}$) nor its complement, then its automorphism group must have “permutation rank 3” (we thank Babai for this observation): the set of orbitals (=orbits on the pairs $V \times V$) must be precisely the diagonal $\{(v, v) : v \in V\}$, the edges, and the non-edges. As the rank-3 permutation groups have a complete classification [Lie87], perhaps that classification would inspire a positive answer to the following question:

Open Question 1.8. Can isomorphism of symmetry-characterized graphs (to one another and/or to arbitrary graphs) be decided in polynomial time?

The above question would be interesting to resolve even under the promise that (one or both of) the graphs are symmetry-characterized. Which also raises the question of: What is the complexity of deciding whether a graph is symmetry-characterized? (The analogous question for polynomials was raised in [Gro12b, Open Question 3.4.8], and is still open as far as we are aware.)

We show, using one algorithm, that finding separating modules and computing multiplicities up to degree d can be done in $n^{O(d)}$ time (Theorems 3.1 and 6.3).

Open Question 1.9. Is there an algorithm to find a separating module in degree $\leq d$ faster than $n^{O(d)}$ time? Is there an algorithm to find a multiplicity obstruction in degree $\leq d$ faster than $n^{O(d)}$ time?

It is not clear to us which of these problems (if either) should be easier: on the one hand, finding a separating module requires just finding a single separating module, not computing how many of that equivalence class vanish. On the other hand, perhaps there are representation-theoretic tricks that allow one to compute a multiplicity obstruction without explicitly finding the underlying separating modules involved. It would also be interesting if there were efficient reductions in either direction between these two problems.

In addition to the result of Dawar, Pago, and Seppelt mentioned above [DPS26]—that the count of H -subgraphs of G (the value of $S_H(G)$) can be computed by symmetric algebraic circuits of size $n^{O(hdtw(H))}$ —this same count can be computed in time $f(H)|G|^{O(tw(H))}$, where tw denotes the treewidth and f is some function that depends only on H [AYZ95]. (Marx [Mar10] showed that this essentially cannot be improved to $f(H)|G|^{o(tw(H)/\log tw(H))}$ unless the Exponential Time Hypothesis is false.) We ask whether the same holds for symmetric algebraic circuits:

Open Question 1.10. Can S_H be computed symbolically by (symmetric) algebraic circuits of size or bit-size $f(H)|G|^{O(tw(H))}$?

Finally, we hope to make it the subject of future work to apply these ideas in the setting of TENSOR ISOMORPHISM (e. g., [GQ23]). There, as in the case of GCT more generally, orbits need no longer be closed, so we are not guaranteed that they are separated by invariants at all, and the use of separating modules in that setting, rather than merely invariants, may be more of a necessity than for GRAPH ISOMORPHISM. As with our study of GI, applying these ideas to TENSOR ISOMORPHISM has the possibility of revealing new insights both about TENSOR ISOMORPHISM itself, and about the representation-theoretic approach to separating orbits (and orbit closures) suggested by GCT.

1.3 Paper outline

In Sec. 2 we give preliminaries. In Sec. 3 we prove that separating modules of degree d can be computed in $n^{O(d)}$ time, their equivalence with invariants of low-degree, and their equivalence with the subgraph census of small subgraphs (=the initial coloring of low-dimensional WL). In Sec. 4 we connect symmetric circuit size of S_n -modules to other complexity measures on the symmetric group [DFL⁺21]. In Sec. 5 we prove the equivalence between separating modules of symmetric circuit size $n^{O(k)}$ and $O(k)$ -WL. In Sec. 6 we prove our results on the multiplicity approach—including limitations of multiplicity and occurrence obstruction, and the proof that multiplicity obstructions are in general stronger than occurrence obstructions. Finally, in Sec. 7 we give the connections between invariant polynomials and the Reconstruction Conjectures.

2 Preliminaries

We will need the following basic lemma.

Lemma 2.1. *For any $1/\log_2 n \leq \varepsilon < 1$ and $s < \varepsilon n^{1-\varepsilon}$, we have $n^s < \binom{n}{r}$ whenever $s/\varepsilon \leq r \leq n/2$.*

By setting ε as an appropriate function of n , this lets us extend the range of s to which the lemma applies almost all the way up to $\Theta(n)$: the best possible is when $\varepsilon = 1/\log_2 n$, for then the lemma applies for s up to $n/(2\log_2 n)$, and $s/\varepsilon = n/2$, so it works only for exactly one value of r , namely $s/\varepsilon = r = n/2$. (And this explains the lower bound on ε : when $\varepsilon < 1/\log_2 n$, the range of r to which the lemma applies becomes empty.)

Proof. When $r \leq \frac{n}{2}$, $\binom{n}{r}^r \leq \binom{n}{r}$. Thus, it suffices to show that $n^s < \binom{n}{r}^r$. Since $\binom{n}{r}$ is an increasing function of r for $r \leq n/2$, we prove the result for $r = s/\varepsilon$:

$$\begin{aligned} n^s < \left(\frac{n}{s/\varepsilon}\right)^{s/\varepsilon} &\iff (s/\varepsilon)^{s/\varepsilon} < n^{s(1/\varepsilon-1)} \\ &\iff (s/\varepsilon)^{1/\varepsilon} < n^{(1-\varepsilon)/\varepsilon}, \end{aligned}$$

and the latter holds iff $s < \varepsilon n^{1-\varepsilon}$, completing the proof of the lemma. \square

We will use the following group theoretic definition throughout this paper.

Definition 2.2. The stabilizer subgroup of G with respect to x is the set of all elements in G that fix x , i. e. $\text{stab}_G(x) = \{g \in G : gx = x\}$.

2.1 Algebraic circuits

By an algebraic circuit over a ring R (which, in this paper, we will typically take to be either the integers or a field) we mean a directed acyclic graph where each source node is labeled either by a variable or by a constant from R , and each non-source node is labeled either by $+$ or \times , which we refer to as addition gates and multiplication gates, respectively. Each gate in an algebraic circuit naturally computes a polynomial over R in the variables occurring on its source nodes, which we refer to as input gates.

By the *size* of an algebraic circuit we mean its number of gates. By the *bit-size* of C , we mean the total bit-size of a natural description of C . For example, when R has a natural notion of bit-size, such as $R \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{Q}(i)\}$, we mean the sum of all the bit-sizes of the constants in C , plus a description of the DAG (say, by an adjacency matrix or adjacency list), plus the labels of all the gates.

A circuit is *constant-free* if the only constants it uses on its source nodes are from $\{0, 1, -1\}$ (other constants can effectively be used by using arithmetic gates to build them up from $\{\pm 1\}$). Over the integers, circuits of total bit-size s and constant-free circuits of size s are essentially equivalent:

Lemma 2.3. *Given a (multi-output) algebraic circuit over \mathbb{Z} , of total bit-size s and depth Δ , there is a constant-free algebraic circuit of size $O(s \log s)$ and depth $\Delta + O(\log s)$ computing the same set of polynomials.*

Conversely, a constant-free algebraic circuit of size $O(s)$ is already an algebraic circuit of total bit-size $O(s^2)$.

Proof. Let C be a multi-output integer circuit of total bit-size s and depth Δ . We will build a new constant-free circuit C' that computes the same set of functions as C .

If one of the source gates of C is labeled by the integer constant c , then in C' that source gate gets replaced by a constant-free gadget computing c from its binary representation. If $c = \sum_{i=1}^k c_i 2^i$ where all c_i are in $\{0, 1\}$, then the gadget first computes 2 using an addition gate $1 + 1$; from 2, it then uses repeated squaring to compute $\{1, 2, 4, 2^3, \dots, 2^k\}$. To keep the depth of this gadget small, it first computes $2, 2^2, 2^4, 2^8, 2^{16}, \dots, 2^{2^i}, \dots, 2^k$ by repeated squaring in depth $\log k$. From these, it then computes the remaining powers 2^i (such that $c_i = 1$) following the usual repeated-squaring-and-multiply scheme. (For example, it computes 2^7 by taking the product $2 \times 2^2 \times 2^4$.) Finally, it replaces the gate labeled by c with the sum of $\{2^i : c_i = 1\}$. The total size of this circuit is $O(k \log k)$ and its depth is $O(\log k)$.

The rest of C' is the same as C .

Thus, the depth of C' might be $O(\log k)$ more (additively) than that of C , where k is the maximum bit-size of any constant used in C . Since C had total bit-size s , we have $k \leq s$.

The total size of C' is larger than C by a factor of $\sum b_i \log b_i$, where b_i are the bit-sizes of the constants used on the source nodes of C . Since the total bit-size of C is at most s , the sum $\sum b_i \log b_i$ is at most $s \log s$. For let $s_0 = \sum b_i \leq s$; then we have:

$$\begin{aligned} \sum b_i \log b_i &= s_0 \cdot \sum (b_i/s_0) \log b_i \\ &= s_0 \cdot \left(\log s_0 + \sum (b_i/s_0) \log(b_i/s_0) \right) \\ &\leq s_0 \log s_0 \\ &\leq s \log s \end{aligned}$$

where the third line follows because $\{b_i/s_0\}$ is a probability distribution, so $\sum (b_i/s_0) \log(b_i/s_0)$ is the negative of the entropy, hence is non-positive.

For the converse, the $O(s^2)$ is just the number of bits needed to describe a DAG on s vertices (plus $O(s)$ to describe the type of each gate). \square

2.2 Representation theory

We use some basic definitions from any standard representation theory textbook, e. g., [FH91, DF04, JK09, Sag01].

Given a group G and a vector space V over a field \mathbb{F} , a *representation* of G on V is a homomorphism $\rho : G \rightarrow \text{GL}(V)$, namely, such that $\rho(g_1 g_2) = \rho(g_1) \rho(g_2)$ for all $g_1, g_2 \in G$. When we think of the vector space V as equipped with such an action of G , we call V a *G -module*. If $\dim_{\mathbb{F}} V = k < \infty$, by choosing a basis for V we can use the alternate notation $\rho : G \rightarrow \text{GL}_k(\mathbb{F})$. We sometimes refer to the vector space V as the G -module if the homomorphism ρ is obvious. Two representations ρ, τ are *equivalent* if their homomorphisms are conjugate, i. e., there exists some invertible matrix A such that $\rho(g) = A \tau(g) A^{-1}$ for all $g \in G$. In other words, representations are equivalent up to change of basis in the vector space. Given a representation (ρ, V) , a *sub-representation* is a subspace W of V that is invariant under the action of G , viz., $\rho(g)W \subseteq W$ for all $g \in G$. An *irreducible representation* or *irrep* is a representation that has no nonzero proper sub-representation.

Maschke's Theorem tells us that for a finite group G , every representation over a field of characteristic zero (or more generally coprime to $|G|$) is equivalent to a direct sum of irreps. The *multiplicity*

of an irrep ρ in a representation σ is the number of times representations equivalent to ρ occur when writing σ as a direct sum of irreps (this number is well-defined).

A linear function between two representations $f: U \rightarrow V$ of the same group G is called *G-equivariant* if $f(g \cdot u) = g \cdot f(u)$ for all $u \in U$. We denote the set of all G -equivariant linear functions $U \rightarrow V$ by $\text{Hom}_G(U, V)$.

If G is a group, a G -set is a set X together with an action of G on X . If $H \leq G$ is a subgroup, then G acts on the set of cosets $G/H := \{gH : g \in G\}$ by left multiplication (every G -set is a disjoint union of such G -sets). Given a G -set X , there is an associated representation denoted $\mathbb{F}[X]$, which has a basis $\{e_x : x \in X\}$ that is in bijection with the set X , and on which g acts by $g \cdot e_x := e_{g^{-1} \cdot x}$ (as in the case of the action on functions, the inverse is needed to make it a left action).

Observation 2.4. Suppose V is a representation of G over a field \mathbb{F} and $H \leq G$. Then

$$\text{Hom}_G(\mathbb{F}[G/H], V) \cong \{v \in V : H \leq \text{stab}_G(v)\},$$

where \cong denotes a bijection of sets.

Proof. Given $f \in \text{Hom}_G(\mathbb{F}[G/H], V)$, let $v = f(H)$. Then for $h \in H$ we have $hv = hf(H) = f(hH) = f(H) = v$, where the middle equality follows by equivariance, and thus $H \leq \text{stab}_G(v)$.

Conversely, given $v \in V$ such that $H \leq \text{stab}_G(v)$, we claim there is a unique G -equivariant linear map $\mathbb{F}[G/H] \rightarrow V$ defined by $gH \mapsto g \cdot v$. Since the cosets of H form a basis of $\mathbb{F}[G/H]$ by construction, the main thing to check is that this function is well-defined. And for the latter, we need to show that if $gH = g'H$ then $gv = g'v$. We have $gH = g'H \Leftrightarrow H = g^{-1}g'H \Leftrightarrow g^{-1}g' \in H$. And in that case, we have $g^{-1}g'v = v$; multiplying both sides by g we get $g'v = gv$ as desired. \square

We will also use the following standard facts about representations of direct products. Given groups G, H and representations $\rho: G \rightarrow \text{GL}(V)$ and $\sigma: H \rightarrow \text{GL}(U)$, we get a representation $(\rho \boxtimes \sigma): G \times H \rightarrow \text{GL}(V \otimes U)$ defined by

$$(\rho \boxtimes \sigma)((g, h)) = \rho(g) \otimes \sigma(h).$$

If the characteristic of \mathbb{F} is coprime to $|G \times H|$, then the set of irreps of $G \times H$, up to equivalence, is precisely $\{\rho \boxtimes \sigma : \rho \text{ an irrep of } G, \sigma \text{ an irrep of } H\}$.

Lastly, given a group G and a subgroup $G_1 \leq G$, if V is a G -module with associated representation $\rho: G \rightarrow \text{GL}(V)$, then $\text{Res}_{G_1}^G V$ denotes the same vector space V , considered as a G_1 -module “by restriction,” that is, the G_1 -representation associated to $\text{Res}_{G_1}^G V$ is precisely $\rho|_{G_1}: G_1 \rightarrow \text{GL}(V)$. A straightforward exercise is that if $H_1 \leq H$, V is a G -module and U is an H -module, then $\text{Res}_{G_1 \times H_1}^{G \times H}(V \boxtimes U) = (\text{Res}_{G_1}^G V) \boxtimes (\text{Res}_{H_1}^H U)$.

2.3 Representations of the symmetric group

We use S_n to denote the symmetric group, consisting of the $n!$ permutations of the set $[n] = \{1, \dots, n\}$. We refer the reader to standard texts [FH91, Ful97, JK09, Sag01] for more details on the representation theory of the symmetric groups S_n .

A partition $\lambda = (\lambda_1, \dots, \lambda_n)$ of n is a list of non-negative integers $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ such that $\sum_{i=1}^n \lambda_i = n$. The irreducible representations of S_n are indexed in a natural way by the partitions of n —there is exactly one irreducible representation associated to each partition λ that we will describe next, and all irreducible representations of S_n (in characteristic zero) arise this way.

We now partially describe the irreducible representation associated to λ , known as a *Specht module* (of shape λ). We first describe a simpler, but non-irreducible representation. A *row tabloid* of shape λ is a list of subsets of $[n]$ that partition $[n]$, and such that the i -th subset has size exactly λ_i . The symmetric group acts transitively on the row tabloids of shape λ , and we define the row tabloid representation as the associated permutation representation.

The Specht module of shape λ is a quotient of the row tabloid representation of shape λ ; we will mostly not need the details of this construction, so we leave the details to the standard texts cited above.

The *character* of a representation $\rho: G \rightarrow \mathrm{GL}(V)$ is the function $\chi_\rho: G \rightarrow \mathbb{F}$ given by $\chi_\rho(g) := \mathrm{tr}(\rho(g))$.

Fact 2.5. The character values of S_n are integers of magnitude at most $\sqrt{n!}$.

Proof. The character values of S_n are all integers (e. g., follows directly from the Frobenius character formula [FH91, Frobenius Formula 4.10]), and since the dot product of a character with itself is $1/n!$ times the sum of the squared norms of the character values, and the dot product of an irreducible character with itself is 1, every character value is strictly less than $\sqrt{n!}$. \square

Fact 2.6. If V is the defining representation of S_n , then $V^{\otimes k}$ contains only representations of S_n corresponding to partitions λ with $\lambda_1 \geq n - k$.

Proof. It is standard that the defining representation is the direct sum of a trivial representation and the so-called standard representation, corresponding to $\lambda = (n - 1, 1)$. Since tensor distributes over direct sum, the trivial representations here have the effect of including (not necessarily with the same multiplicity) the irreps from $V^{\otimes k'}$ with $k' \leq k$ among those in $V^{\otimes k}$. But otherwise they do not generate any “new” irreps that aren’t already present in some tensor power of the standard representation.

For the tensor powers of the standard representation, see, for example, [GC05, Prop. 1], which gives a fairly explicit combinatorial formula for the multiplicities of the representations appearing. It is apparent from that formula that only representations with $\lambda_1 \geq n - k$ can appear. \square

Corollary 2.7. Let $\mathbb{F}[X]$ be the polynomial ring in n^2 variables x_{ij} , arranged into an $n \times n$ matrix X , acted on by S_n by conjugation by ordinary permutation matrices (equivalently, $\pi \cdot x_{ij} = x_{\pi^{-1}(i), \pi^{-1}(j)}$). Then among the degree- d polynomials in this ring, the S_n representations that occur all correspond to partitions λ with $\lambda_1 \geq n - 2d$.

Proof. Let V denote the defining n -dimensional representation of S_n ; then as an S_n -representation, the degree-1 polynomials are $V \otimes V$. Thus the degree- d polynomials are the d -th symmetric power $\mathrm{Sym}^d(V \otimes V)$, which is contained in $V^{\otimes 2d}$. The result now follows from Fact 2.6. \square

The following result will only be used in a kind of “side result” at the end of Section 5.2.

Lemma 2.8. Let $\mathbb{F}[X]$ be the polynomial ring in n^2 variables x_{ij} arranged into an $n \times n$ matrix, acted on by S_n as above. Let $I_{bool} := \langle x_{ij}^2 - x_{ij} : i, j \in [n] \rangle$. Let $\pi: \mathbb{F}[X] \rightarrow \mathbb{F}[X]/I_{bool}$ be the natural quotient map.

If V is an irreducible representation contained in $\mathbb{F}[X]$ that contains some $f \in V$ such that there exists a $\{0, 1\}$ -matrix A with $f(A) \neq 0$, then V is not contained in I_{bool} and $\pi(V)$ is equivalent to V as S_n -representations.

It follows that if $V \leq \mathbb{F}[X]$ is any test module (not necessarily irreducible) that is not identically zero on all $\{0, 1\}$ matrices, then $\pi(V) \leq \mathbb{F}[X]/I_{bool}$ is a non-zero test module.

Proof of Lemma 2.8. First, I_{bool} is a sub-representation of S_n : if $f \in I_{bool}$, then $f = \sum_{i,j} f_{ij} \cdot (x_{ij}^2 - x_{ij})$ for some polynomials f_{ij} (these f_{ij} will not in general be uniquely determined by f , but that non-uniqueness will not be an issue here). Then for $\sigma \in S_n$, since $\sigma \cdot (x_{ij}^2 - x_{ij})$ is another polynomial of the same form, namely $x_{k\ell}^2 - x_{k\ell}$ where $\sigma(k) = i$ and $\sigma(\ell) = j$, we have $\sigma \cdot f = \sum_{i,j} f_{ij} \cdot (x_{\sigma^{-1}(i), \sigma^{-1}(j)}^2 - x_{\sigma^{-1}(i), \sigma^{-1}(j)})$, so σf is also in I_{bool} .

Thus $V \cap I_{bool}$ is also a representation of S_n . Since V is irreducible, we must have $V \cap I_{bool}$ is either 0 or all of V .

Finally, it remains to show that if the polynomials in V do not all vanish identically on all $\{0, 1\}$ matrices, then V is not contained in I_{bool} . For then it follows from the above that $V \cap I_{bool} = 0$, and thus π maps V isomorphically onto its image $\pi(V)$ (isomorphism of S_n -representations).

To show the latter, it suffices to show that I_{bool} is a radical ideal. Recall that an ideal I is *radical* if $f^k \in I$ for some $k \geq 1$ implies that $f \in I$. Hilbert's Nullstellensatz then tells us that if f vanishes on all $\{0, 1\}$ matrices, then some power of f must be in I_{bool} , so if we show that I_{bool} is radical, it follows that any f vanishes on all $\{0, 1\}$ matrices if and only if it is in I_{bool} .

This fact follows from some standard commutative algebra, but we include the proof here for completeness.

Since each generator of I_{bool} is on a different variable, we have $\mathbb{F}[X]/I_{bool} \cong \bigotimes_{i,j} \mathbb{F}[x_{ij}]/\langle x_{ij}^2 - x_{ij} \rangle$. From the (generalized) Chinese Remainder Theorem, we have that $\mathbb{F}[x]/\langle x^2 - x \rangle \cong (\mathbb{F}[x]/\langle x \rangle) \oplus (\mathbb{F}[x]/\langle x - 1 \rangle) \cong \mathbb{F} \times \mathbb{F}$. Thus $\mathbb{F}[X]/I_{bool}$ is a tensor product over \mathbb{F} of direct products of copies of \mathbb{F} ; the result is a direct product of 2^{n^2} copies of \mathbb{F} .

But if $f^k \in I_{bool}$, then the image of f modulo I_{bool} is nilpotent $f^k \equiv 0 \pmod{I_{bool}}$. But since $\mathbb{F}[X]/I_{bool}$ is a direct product of fields, the only nilpotent element there is 0 itself, thus $f \equiv 0 \pmod{I_{bool}}$, in other words $f \in I_{bool}$. This completes the proof that I_{bool} is radical, and thus the proof of the lemma. \square

3 Degree of separating modules, invariants, and subgraph census

In this section, we show that degree d separating modules can be found in $n^{O(d)}$ time, and are equivalent in power to degree- $\Theta(d)$ invariants. We introduce the notion of “support-degree” which counts how many vertices are “involved” in a monomial (rather than degree, which counts edges), and show that separating modules of support-degree $\Theta(d)$, invariants of support-degree $\Theta(d)$, and the census of subgraphs on $\Theta(d)$ vertices (the initial coloring of $\Theta(d)$ -WL) are all equivalent. Since the algorithmic result helps motivate the exploration of degree in the first place, we start there.

3.1 Algorithms for finding low-degree separating modules

Theorem 3.1. *There is an algorithm that, given two n -vertex graphs G, H and an integer d between 0 and n^2 , finds a separating module of degree $\leq d$ for G, H or reports that none exists, in time $n^{O(d)}$.*

When we say “computes a separating module”, we mean that the algorithm outputs a list of integer polynomials, each specified by the list of monomials and nonzero coefficients, such that those polynomials span a separating module.

We note here that even if one is working with polynomials over the complex numbers, the algorithm works in the usual Turing machine model: A consequence of the proof is that, if there is such a separating module, then there is one that has a basis consisting of polynomials with integer coefficients of bit-size at most $n^{O(d)}$.

(The same result holds with the bound $2^{O(d^2)} n^{O(d)}$ when using the notion of “support-degree” that we introduce in Definition 3.3 below. The proof is the same, *mutatis mutandis*.)

As a warm-up that involves essentially no representation theory, we start by proving the statement for invariants only:

Proposition 3.2. *There is an algorithm that, given two n -vertex graphs G, H and an integer d between 0 and n^2 , finds a separating invariant of degree $\leq d$ for G, H or reports that none exists, in time $n^{O(d)}$.*

Proof. Let R be the polynomial ring modulo the Boolean axioms, that is, $R = \mathbb{F}[x_{ij} : i, j \in [n]]/\langle x_{ij}^2 - x_{ij} : \forall i, j \rangle$, and let $R_{\leq d}$ be the vector subspace consisting of all (multilinear) polynomials

of degree at most d . (In fact, our proof will work even if we replace R with the un-quotiented polynomial ring $\mathbb{F}[x_{ij}]$, but we will not need that here.) Let $N = \binom{n}{2}$ be the total number of variables (the proof works for other variants of graphs *mutatis mutandis* by changing N).

For each of the $\leq N^{O(d)} = n^{O(d)}$ monomials of degree $\leq d$, let f be the invariant polynomial which is the sum over the S_n -orbit of that monomial. If $f(G) \neq f(H)$, then let $\hat{f}(X) := f(X) - f(G)$. We then have that \hat{f} is an invariant of degree $\leq d$ such that $\hat{f}(G) = 0$ and $\hat{f}(H) \neq 0$.

We finish with the runtime analysis. Enumerating the monomials takes $n^{O(d)}$ time, and evaluating f on G and H takes $n^{O(d)}$ time. The naive way of computing f by summing over its S_n orbit would take $n!$ time, so we employ a different approach: each degree- d monomial $x_{i_1 j_1} \cdots x_{i_d j_d}$ corresponds to a d -edge graph M with vertex set $[n]$ and edge set $\{(i_1, j_1), \dots, (i_d, j_d)\}$. The sum over monomials can be computed by enumerating all labeled graphs isomorphic to M and summing over the monomials corresponding to those graphs. Let $s = |\{i_1, \dots, i_d, j_1, \dots, j_d\}|$, which is in between d and $2d$. Then M has at least $n - s \geq n - 2d$ isolated vertices. We enumerate the labeled graphs isomorphic to M by first picking the s non-isolated vertices—of which there are $\binom{n}{s} \leq n^{O(d)}$ many—and then, on those s vertices, brute-force enumerating all $s!$ permutations applied to the vertices of M , maintaining the set of labeled graphs that occur. (This may seem like too much time, but if $s!$ is approaching $n!$, then d must be approaching n , so the goal of $n^{O(d)}$ is a very generous bound.) The total runtime of this procedure is then $s!n^{O(d)} \leq 2^s \log s n^{O(d)} \leq 2^{2d} \log(2d) n^{O(d)} \leq n^{2d \frac{\log 2d}{\log n}} n^{O(d)} \leq n^{4d(1+o(1))+O(d)} \leq n^{O(d)}$, as claimed. \square

We now come to the proof of the more general theorem for separating modules, which starts from the same basic idea but needs more (algorithmic) representation theory.

Proof of Theorem 3.1. Let R be the polynomial ring modulo the Boolean axioms, that is, $R = \mathbb{F}[x_{ij} : i, j \in [n]] / \langle x_{ij}^2 - x_{ij} : \forall i, j \rangle$, and let $R_{\leq d}$ be the vector subspace consisting of all (multilinear) polynomials of degree at most d . (In fact, our proof will work even if we replace R with the un-quotiented polynomial ring $\mathbb{F}[x_{ij}]$, but we will not need that here.)

Let π_1, π_2 be two generators of the symmetric group S_n , say $\pi_1 = (12), \pi_2 = (12 \cdots n)$. We can compute the action of π_i on (multilinear) monomials of degree $\leq d$ to get a matrix M_i where for a monomial m , we have $M_i(m) = \pi_i(m)$. Since S_n merely permutes monomials, the matrices M_i are $\{0, 1\}$ -valued (in fact, permutation matrices), and they have dimension at most $\dim R_{\leq d} \leq \binom{n^2+d}{d}$. So this can be computed in time at most $\binom{n^2+d}{d} \cdot n^c \leq n^{O(d)}$ (for some absolute constant c , to account for bookkeeping overhead; here we also use the assumption that $d \leq n^2$ to get that $\binom{n^2+d}{d} \leq \binom{2n^2}{d} \leq (2n^2)^d \leq n^{d(2+\log_2 2)} \leq n^{O(d)}$).

Now we apply the algorithm of Chistov, Ivanyos, and Karpinski [CIK97, Theorem 6] to the pair M_1, M_2 , to break up $R_{\leq d}$ into irreps. That algorithm breaks a representation up into a direct sum of absolutely irreducible modules over the algebraic closure of the ground field; however, in the case of S_n , irreducible representations of S_n over any field of characteristic zero are absolutely irreducible (meaning, they remain irreducible over the algebraic closure), thus their algorithm indeed returns a decomposition into irreps. That means we get a basis for $R_{\leq d}$ that is partitioned into bases for irreps. We can also use Chistov–Ivanyos–Karpinski [CIK97, Corollary 3] or Brooksbank–Luks [BL08, Theorem 3.5] to determine which irreps appearing are isomorphic to which other ones, as well as find an isomorphism between them. Thus, for each isomorphism type of irrep, at this point we have constructed a basis $v_1, \dots, v_k \in R_{\leq d}$ of one copy of that irrep, and linear isomorphisms between the copies of the irreps. So now focus on an isotypic component—the direct sum of all irreps that are equivalent to a given irrep—say it contains m copies of a single irrep. Let $v_{i,j}$ be the image of v_i in the j -th isomorphic copy, that is, for all j, j' , the map $v_{i,j'} \mapsto v_{i,j}$ for all $i = 1, \dots, k$ is an equivalence of S_n -representations.

Now, for each of the m copies in our isotypic component, $j = 1, \dots, m$, consider the vector $u_j := (v_{1,j}(G), \dots, v_{k,j}(G))$ (here remember our vectors v_{ij} are actually polynomials in R , so by $v_{i,j}(G)$ we mean apply that polynomial to the adjacency matrix of G). Arrange those vectors u_j as

the columns of a matrix U , of size $k \times m$, where k is the dimension of the irrep we are focusing on and m is its multiplicity in $R_{\leq d}$. Then the kernel of U , $\{x : Ux = 0\}$, consists of linear combinations of the columns that are zero on G . Each such linear combination corresponds to one copy of the irrep in that isotypic component that vanishes on G .

By iterating over all the isotypic components that occur (of which there can be at most $\dim R_{\leq d} \leq n^{O(d)}$), we can then find all test modules that vanish on G . By doing similarly for H , we can then find all separating modules in $R_{\leq d}$.

Finally, to turn the rational polynomials into integer ones we can clear denominators. \square

3.2 Relationship between separating modules and separating invariants

With that motivating theorem in hand, we now pursue the equivalence between bounded-degree separating modules, bounded-degree separating invariants, and the counts of bounded-order subgraphs (the initial coloring of $O(1)$ -WL).

Because our polynomials have variables for the edges, but WL works on vertices, in our work we have often run into a sort of “quadratic mismatch” between the two (a monomial of degree d can involve up to $2d$ edges, and conversely a d -tuple of vertices can have up to $\Theta(d^2)$ edges, which would be encoded by a monomial of degree $\Theta(d^2)$). We introduce here a notion of “support-degree” for monomials that counts the number of vertices involved in a monomial:

Definition 3.3 (Support-degree). The *support-degree* of a monomial $x_{i_1 j_1} \cdots x_{i_d j_d}$ is

$$\text{supp-deg}(x_{i_1 j_1} \cdots x_{i_d j_d}) := |\{i_1, \dots, i_d, j_1, \dots, j_d\}|.$$

The support-degree of a polynomial f , denoted $\text{supp-deg}(f)$, is the maximum of the support-degree of any of its monomials. If T is a set of polynomials (in particular, a test module), we define its support-degree to be $\text{supp-deg}(T) := \max_{f \in T} \text{supp-deg}(f)$.

We always have

$$\text{supp-deg}(f) \leq 2 \deg(f) \leq O(\text{supp-deg}(f)^2).$$

Observation 3.4. For all polynomials f, g ,

$$\text{supp-deg}(fg) \leq \text{supp-deg}(f) + \text{supp-deg}(g),$$

and strict inequality is possible. That is, support-degree gives a filtration, but not a grading, on the polynomial ring in variables x_{ij} , as well as its quotient by the Boolean equations $x_{ij}^2 - x_{ij}$.

Proof. For two monomials $m = x_{i_1 j_1} x_{i_2 j_2} \cdots x_{i_d j_d}$ and $m' = x_{\ell_1 k_1} \cdots x_{\ell_e k_e}$ supported on vertex sets $V = \{i_1, \dots, i_d, j_1, \dots, j_d\}$ and $V' = \{\ell_1, \dots, \ell_e, k_1, \dots, k_e\}$, respectively, we have

$$\begin{aligned} \text{supp-deg}(mm') &= |V \cup V'| = |V| + |V'| - |V \cap V'| \\ &= \text{supp-deg}(m) + \text{supp-deg}(m') - |V \cap V'| \\ &\leq \text{supp-deg}(m) + \text{supp-deg}(m') \end{aligned}$$

Since $V \cap V'$ need not be empty (e. g. $m = x_{12}, m' = x_{13}$), the final inequality can be strict. \square

We next recall some key definitions from [DW25].

Definition 3.5. If G is a permutation group acting on the variables of a circuit C , we say C is G -symmetric if every $g \in G$ extends to at least one automorphism of C .

When $G \leq S_{n^2}$ is the action of S_n on ordered pairs $\pi \cdot x_{i,j} = x_{\pi^{-1}(i), \pi^{-1}(j)}$ we call C *square-symmetric*.

Remark 3.6. We believe all the results in our paper continue to work *mutatis mutandis* in the settings of (a) undirected graphs, where x_{ij} and x_{ji} are now considered the same variable, and the action is of S_n by $\pi \cdot x_{ij} = x_{\pi^{-1}(i)\pi^{-1}(j)}$, (b) bipartite graphs where $S_n \times S_m$ acts on the variables by $(\pi, \rho) \cdot x_{ij} = x_{\pi^{-1}(i)\rho^{-1}(j)}$, (c) with or without self-loops in any of these situations; and many other standard variants of graphs.

Definition 3.7. If G is a permutation group acting on the variables of a circuit C , we say C is *(G-)rigid* if any permutation of the inputs that comes from G extends to at most one automorphism of C .

Our next lemma, which has several uses, is to show that irreducible separating modules V of degree d can be computed by symmetric algebraic circuits of size $n^{O(d)}$, in the sense that there is such a circuit with multiple outputs, whose outputs form a spanning set for V .

We note that, although it may seem “obvious” that given a separating module of degree d , it should have a symmetric circuit of size $n^{O(d)}$ —since, after all, there are only $n^{O(d)}$ monomials total of that degree—there are in fact several obstacles to proving this. One challenge is that we really want to bound the total bit-size of the circuits and not just the number of gates—later this will be helpful in order to translate from algebraic circuits back to Boolean threshold circuits. Another obstacle is that, even if we can find a single f with small coefficients inside a given separating module, and compute f by a (non-symmetric) depth-2 algebraic circuit of size $n^{O(d)}$, to get a *symmetric* circuit, the natural approach is then to take the orbit of f . But if we did not choose f carefully, its orbit could have size $n!$ (in fact, “most” f will have orbits of that, maximum, size). When d is small, the elements of that orbit cannot be linearly independent, but the naive way of getting a symmetric circuit from f cannot avoid computing all the elements in the orbit of f . So we must also identify an element of the separating module whose orbit has size $n^{O(d)}$ (and not merely spans a set of dimension $n^{O(d)}$, which is always true in degree d). To overcome all of these obstacles simultaneously we use quite a bit of information from the representation theory of the symmetric group (albeit mostly in a black-box manner), in the proof of the following key lemma.

Lemma 3.8. *Suppose two graphs G, H are separated by separating modules in degree $\leq d$. Then there is an irreducible separating module V in degree $\leq d$ such that there is a rigid square-symmetric multi-output circuit C of total bit-size $n^{O(d)}$ and depth 2 whose outputs are a spanning set for V .*

If we replace degree with support-degree, the same result holds with total bit-size $2^{O(d^2)}n^{O(d)}$.

Proof. The proof for support-degree is essentially the same as that for ordinary degree, *mutatis mutandis*; the main difference is that there are $2^{O(d^2)}\binom{n}{d}$ monomials of support-degree d rather than $n^{O(d)}$ monomials of degree d . Since the proof for support-degree is otherwise the same, we only give the details of the proof in the case of ordinary degree.

In the first half of the proof we find a polynomial v_0 whose orbit separates G from H , has orbit size $n^{O(d)}$, and whose coefficients (in the monomial basis) have bit-size $n^{O(d)}$. In the second half we then use that polynomial v_0 to build a circuit satisfying the statement of the lemma, whose outputs span V .

Finding v_0 . Towards that end, suppose there is a separating module in degree d that vanishes on G but not on H . By the proof of Theorem 3.1, we can compute bases for all the separating modules in $\mathbb{F}[X]_{\leq d}$, consisting of polynomials with integer coefficients of bit-size at most $n^{O(d)}$. (Note that this is not yet enough since it is possible that all of those polynomials have orbits as large as $n!$, which is too large for our use.)

Let V be one such irreducible separating module, and let λ be the partition indexing the isomorphism type of V . Let $S_\lambda := S_{\lambda_1} \times \cdots \times S_{\lambda_n}$ be the corresponding Young subgroup. By, e.g., [Ful97, Sec. 7.2], V has a surjective S_n -equivariant linear map from $\mathbb{F}[S_n/S_\lambda]$, and thus by Obs. 2.4 V contains a non-zero vector that is stabilized by S_λ (its stabilizer may be larger, but it is at least

stabilized by S_λ). Let $\pi_{inv} = \sum_{\sigma \in S_\lambda} \sigma$ be the (scaled) projection operator onto the S_λ -invariants. Since V contains S_λ -invariants, at least one of its basis vectors v as computed above must have non-zero projection $\pi_{inv}(v)$; we may check all of them, since there are at most $n^{O(d)}$ of them. Let $v_0 = \pi_{inv}(v)$ for some such v . Since v_0 is a sum of at most $|S_\lambda| \leq n!$ many permuted copies of v , it has integer coefficients of magnitude at most $n! \cdot 2^{n^{O(d)}} \lesssim 2^{n^{O(d)} + n \log n} \leq 2^{n^{O(d)}}$, hence bit-size at most $n^{O(d)}$. We have thus found the v_0 we sought at the outset.

Building the circuit. The remainder of the proof is now to build our circuit based on the polynomial v_0 .

Let C_0 be a depth-2 circuit computing v_0 as a sum of monomials with integer coefficients. For each $\tau \in S_n$, let $\tau \cdot C_0$ be the circuit C_0 applied to τ applied to the input variables x_{ij} . For $\tau \in S_\lambda$, since v_0 is fixed by S_λ , we have that τ must permute the gates of C_0 (which correspond to monomials of v_0) amongst themselves, giving back the same circuit C_0 . The orbit of C_0 thus consists of at most (remember that the stabilizer of v_0 could be larger than S_λ) $|S_n|/|S_\lambda| = n!/\prod_{i=1}^n \lambda_i!$ distinct circuits; let C be the union of these circuits. (Conceptually, we may start with C as a disjoint union, but if two gates compute the same monomial we may merge them together into a single gate.) Then C is a multi-output, square-symmetric circuit whose outputs are precisely the polynomials in the S_n -orbit of v_0 .

Since V is irreducible, the orbit of any nonzero vector in V is a spanning set for V , and thus the outputs of C are a spanning set for V .

Next we show that C is rigid: suppose α is an automorphism of C . If it moves one of the product gates at depth 1, then α moves the corresponding monomials. Since they were distinct monomials, it must also swap the variables that go into those monomials, and thus α cannot act trivially on the variables. Next, suppose instead that α moves some (output) gates at depth 2. Even if two output gates are supported on the same set of monomials, they cannot have the same coefficients, and thus α must also move some of the gates at depth 1 (moving the monomials) or depth 0 (which also has the effect of swapping which monomial each depth-1 gate computes). In either case, α does not fix the variables, and thus C is rigid.

We will now make use of the fact that by Cor. 2.7, λ has $\lambda_1 \geq n - 2d$. Thus $|S_\lambda| \geq (n - 2d)!$, so the number of outputs of C (the orbit of v_0) is at most $n!/(n - 2d)! < n^{2d}$. Since v_0 has degree d , the size of C_0 is at most $n^{O(d)}$, and thus the size of C is at most $n^{2d} \cdot n^{O(d)} \leq n^{O(d)}$. Furthermore, the coefficients appearing in C are the same as those appearing in C_0 , so each coefficient has magnitude at most $2^{O(dn^d \log n)}$, and thus bit-size at most $O(dn^d \log n)$. Since there are most $n^{O(d)}$ such coefficients, the total bit-size of C is at most $n^{O(d)}$. \square

Proposition 3.9. *1. If two graphs G, H are separated by separating modules in degree $\leq d$, then they are separated by an invariant of degree $\leq 2d$. Moreover, this invariant can be computed by a rigid square symmetric algebraic circuit of total bit-size $n^{O(d)}$ and depth 2.*

2. The same holds with degree replaced everywhere with support-degree, with the resulting total bit-size $2^{O(d^2)} n^{O(d)}$.

3. If G, H are separated by a separating module with a spanning set computed by a (rigid) square-symmetric circuit with s gates of total bit-size b , then they are separated by an invariant computed by a (rigid) square-symmetric circuit of with $\leq 2s + 1$ gates of total bit-size $O(b + s \log s)$.

Since separating invariants are a special case of separating modules, this shows that whether measured by degree or by circuit-size, asymptotically there is little difference in the distinguishing power of separating modules and separating invariants for GRAPH ISOMORPHISM.

Proof. 1. If G, H admit a separating module in degree d , let V be an irreducible separating module in degree d . To make the separation of ideas clear, we first prove the existence of an invariant of low degree that also separates G from H , then we show how to reduce its circuit complexity.

Take any nonzero polynomial $f \in V$ with integer coefficients; let f_1, \dots, f_m be its orbit. Let $\hat{f} = \sum_{i=1}^m f_i^2$. Since S_n acts on these irreps by rational matrices, all the f_i have rational coefficients. Thus their squares are polynomials that only take non-negative values on $\{0, 1\}$ -adjacency matrices (or indeed on any real-valued matrices). Thus the output of \hat{f} has the property that it is 0 on an adjacency matrix A if and only if every polynomial f_i vanishes on A , since when any term is non-zero, it must be positive, so the terms can never cancel one another. Since V is irreducible, any nonzero orbit spans V , and thus $\hat{f}(A) = 0$ iff $h(A) = 0$ for all $h \in V$. Thus, if V vanishes on G but not on H , the same is true of \hat{f} .

However, the size of f 's orbit could be as large as $n!$, far too large for our circuit size bound. We know how to choose f so that we get a small circuit.

Let C be the rigid square-symmetric circuit of bit-size $n^{O(d)}$ and depth 2 computing a spanning set for V from Lemma 3.8. Let C' be the circuit that starts the same as C , but then takes each output of C , squares it, and adds them all together. Then $\text{size}(C') \leq 2\text{size}(C) + 1$, and the degree of the polynomial computed by C' is $2\deg(V)$. Furthermore, since C was (square-)symmetric, summing the squares of its outputs produces an invariant polynomial. The same argument as above shows that C' vanishes on an adjacency matrix A if and only if every polynomial in V does.

Finally, to get C' down to depth 2, we note that we can replace each output gate g of C with a depth-2 circuit computing g^2 as a sum of monomials, and of size at most the square of the size of the subcircuit rooted at g . Once this is done, the final addition gate of C' can be merged into the addition gates that are the outputs of C , resulting overall in a circuit whose size is still $n^{O(d)}$ but whose depth is now only 2, that computes a separating *invariant* for G, H of degree at most $2d$, as claimed.

2. The same proof works for support-degree, *mutatis mutandis*.

3. Use the same ‘‘sum the squares of the outputs’’ trick from the first paragraph above. (The extra $\log s$ factor in the bit size is just that the connections on the new squaring gates each take $O(\log s)$ bits to specify.) \square

3.3 Equivalence between support-degree- d separating invariants and d -vertex subgraph counts

In order to characterize the power of invariants of degree d (and hence, by Proposition 3.9, also separating modules of degree $O(d)$), we give an explicit basis for such invariants. If f is an invariant polynomial on the adjacency matrices of n -vertex graphs, consider one of its monomials αm , where $\alpha \in \mathbb{F}$ and m is a product of variables. Let H be the n -vertex graph whose edge set is precisely $E(H) := \{(i, j) : x_{ij} \text{ divides } m\}$. Then we see that for any graph G , $m(G) = 1$ if and only if H occurs in G as a (not necessarily induced) subgraph in the position specified by H . Furthermore, since f is invariant, every monomial of the form $\sigma(m)$ ($\sigma \in S_n$) must also occur in f with the same coefficient α . This leads us to define:

$$S_H := \frac{1}{|\text{Aut}(H)|} \sum_{\sigma \in S_n} \sigma \left(\prod_{(i,j) \in E(H)} x_{ij} \right) = \sum \left\{ \sigma \left(\prod_{(i,j) \in E(H)} x_{ij} \right) : \sigma \in S_n \right\}$$

The normalizing factor here is set so that S_H is precisely the sum over the monomials $\sigma(m)$, with each one occurring with coefficient 1. Put another way: since we are summing over the whole symmetric group, if the graph H associated to m has some automorphisms, we would get each such monomial with coefficient $|\text{Aut}(H)|$. The degree of S_H is $|E(H)|$ and the support-degree of S_H is the number of non-isolated vertices of H .

The preceding reasoning leads quickly to:

Observation 3.10. Every (degree d homogeneous, resp., support-degree d homogeneous) invariant polynomial is a linear combination of the S_H polynomials (where $|E(H)| = d$, resp., $|V(H)| = d$).

Proof. If f is an invariant polynomial, then it is a linear combination of orbits of monomials. To each monomial $m = x_{i_1 j_1} x_{i_2 j_2} \cdots x_{i_k j_k}$, we associate the graph H_m on vertices $\{i_1, i_2, \dots, i_k, j_1, \dots, j_k\}$ with edge set $\{(i_1, j_1), \dots, (i_k, j_k)\}$. Then the sum over the orbit of m is a scalar multiple of S_H . Moreover, if m has degree d , then H_m has exactly d edges, and if m has support-degree d , then H_m has exactly d vertices. \square

We now have all the ingredients in place for the main complexity theorem of this section. To state it more succinctly, we introduce the notation:

- $WLDim_0(G, H)$ is the minimum k such that k -vertex subgraph census (the 0th round of k -WL) distinguishes G and H
- $SepDeg(G, H)$ is the minimum degree of any separating module for G, H ; $SepSuppDeg(G, H)$ is the minimum support-degree of any separating module for G, H .
- $InvDeg(G, H)$ is the minimum degree of any separating invariant for G, H ; $InvSuppDeg(G, H)$ is the minimum support-degree of any separating invariant for G, H .

Theorem 3.11. *For all graphs G, H we have*

$$WLDim_0(G, H) \leq InvSuppDeg(G, H) \leq 2SepSuppDeg(G, H) \leq 2WLDim_0(G, H)$$

and

$$WLDim_0(G, H) \leq 2InvDeg(G, H) \leq 4SepDeg(G, H) \leq O(WLDim_0(G, H))^2.$$

Consequently, $WLDim_0 = \Theta(InvSuppDeg) = \Theta(SepSuppDeg)$, and $WLDim_0$ is $O(1)$ if and only if $InvDeg$ and $SepDeg$ are.

Proof. By Proposition 3.9, if G, H admit a separating module of support-degree (resp. degree) d , then they admit a separating invariant of support-degree (resp. degree) $2d$. This gives the middle inequality of both sequences of inequalities.

By Observation 3.10, invariants of support-degree (resp., degree) d are linear combinations of the polynomials S_K for graphs K on $\leq d$ vertices (resp., edges). The value of those invariants on a graph G are completely determined by the counts of induced subgraphs on $\leq d$ (resp., $2d$) vertices, giving the first inequality of both sequences.

Conversely, the initial coloring of d -WL is determined by the invariants S_K for the graphs K on d vertices, hence is determined by invariants of support-degree $\leq d$ (resp., degree $\leq O(d^2)$), giving the final inequality of both sequences. \square

4 Support size and complexity measures on the symmetric group

A key concept that will be useful in our proofs is that of “support” of an element of a set that is acted on by S_n . In this section we recall the definition of support and some of its basic properties, and we relate it to other complexity measures on the symmetric groups. The notion of support will be used crucially in the next section as well.

Definition 4.1 (Support, cf. [BGS99, DW22]⁷). Suppose S_n acts on a set Ω . A *support* of $\omega \in \Omega$ is a subset $X \subseteq [n]$ (sic!) such that every element in the pointwise stabilizer of X also stabilizes ω , that is

$$\bigcap_{x \in X} \text{stab}_{S_n}(x) \subseteq \text{stab}_{S_n}(\omega).$$

For concreteness, it may help the reader to note that if $X = \{1, \dots, k\}$, then the pointwise stabilizer on the left-hand side of the above definition is merely a copy of S_{n-k} , and the defining condition becomes $S_{n-k} \subseteq \text{stab}_{S_n}(\omega)$ (though, despite the latter notation, it often matters *which* copy of S_{n-k} is being referred to).

Note that we have (at least) two actions of S_n here: the action on Ω , and the defining action on $[n]$. And supports are defined to always be subsets of $[n]$. A typical application of this in our paper will have $\Omega = [n] \times [n]$ the possible edge set of directed graphs on vertex set $[n]$ (or $\Omega = \binom{[n]}{2}$ for undirected graphs), and in that case supports are (by definition) sets of vertices.

Although we don't technically need the following lemma, we find it useful for gaining some intuition about supports, which may aid the reader in following our subsequent proofs.

Lemma 4.2 ([BGS99, DW22]). *Suppose $\omega \in \Omega$ has a support of size less than $n/2$. Then there exists a unique minimum-size support of ω . Moreover, this support is a subset of each support of ω of size $n/2$.*

Definition 4.3 (Support size). Suppose S_n acts on Ω ; the *support size* of $\omega \in \Omega$ is the minimum size of any support of ω . The *support size* of the S_n -set Ω is $\max_{\omega \in \Omega} \text{support-size}(\omega)$.

We will frequently apply this to the case where Ω is the set of gates of a circuit that is acted on by S_n .

Note that if m is a monomial then $\text{supp-deg}(m)$ as in the previous section agrees with the size of a minimal support of m in the sense of Definition 4.1; however, for a more general polynomial f , $\text{supp-deg}(f)$ may not agree with the minimum size of a support for f (it is, instead, the maximum over monomials of the minimum support size of those monomials).

Using ideas from the proof of Lemma 3.8, we now show that the support size of a circuit computing a spanning set for a representation inside a polynomial ring R on which S_n acts is closely related to other complexity measures on the symmetric group. In particular, [DFL⁺21] show that the number $n - \lambda_1$ associated to a partition λ indexing an irrep is closely related to a number of other measures of complexity on the symmetric group, including sensitivity, block-sensitivity, and decision tree complexity. In their setting, they consider irreps inside the coordinate ring (or group algebra) of S_n directly; Proposition 4.4 generalizes the connection to the action of S_n on other polynomial rings.

Proposition 4.4. *Suppose S_n acts on a set Ω , and let R be the polynomial ring on variable set Ω , and let s_Ω denote the support size of Ω (Definition 4.3).*

If $V \subseteq R$ is an irreducible representation of shape λ with $k \geq n - \lambda_1$, occurring in $R_{\leq d}$ then there is a rigid square-symmetric algebraic circuit C of support size at most $\max\{k, d \cdot s_\Omega\}$ computing a spanning set for V .

Conversely, if C is a rigid, symmetric algebraic circuit of support size k computing polynomials in R , then the outputs of C span a representation of S_n , and that representation is a direct sum of irreducible representations of shapes λ with $k \geq n - \lambda_1$.

Proof. (\Rightarrow) (The proof of this direction is very similar to the proof of Lemma 3.8, except now we need not worry about the bit-size of our coefficients any more.) As in Lemma 3.8, V contains a vector

⁷Dawar and Wilsenach [DW22] define the support of a subgroup $G \leq S_n$ rather than of an element $\omega \in \Omega$, but these are equivalent concepts: For $\omega \in \Omega$, if we define $G := \text{stab}_{S_n}(\omega)$ then supports of G in their sense are the same as supports of ω in our sense. And conversely, given $G \leq S_n$, we may take $\Omega = S_n/G$ to be the coset space, and $\omega = G$ to be the trivial coset, and then supports of ω in our sense are the same as supports of G in their sense.

v_0 that is stabilized by the Young subgroup S_λ . As there, let C_0 be a depth-2 circuit computing v_0 as a sum of monomials, and let C be the union of the circuits τC_0 for all $\tau \in S_n$ (omitting repeats). Now, while the number of copies of C_0 is still at most $|S_n|/|S_\lambda| \leq n!/(n-k)! < n^k$, unlike the above proof we no longer know that C_0 itself has size $n^{O(k)}$, so we instead must argue more directly to bound the support size of C .

Now, S_λ contains S_{λ_1} , and since S_λ fixes the output gate corresponding to v_0 , S_{λ_1} does as well. Thus v_0 has a support of size at most $n - \lambda_1 \leq k$. Since the output gates are all in the orbit of v_0 , the same holds for them as well.

Now consider a monomial gate in C . Since it has degree d , it is a product of at most d many variables x_ω , say those corresponding to $\omega_1, \dots, \omega_d \in \Omega$. Since each ω_i has a support of size $\leq s_\Omega$, if a permutation fixes a support for each ω_i , then it fixes the monomial gate. Thus there is a set of size at most ds_Ω such that any permutation fixing that set pointwise fixes the monomial gate. Thus it has a support of size at most ds_Ω .

(\Leftarrow) Conversely, suppose C is a rigid symmetric algebraic circuit of support size s computing a polynomial in R . Let its outputs be f_1, \dots, f_m . Then since C is a rigid symmetric circuit, the action of S_n on the variables permutes the set of output gates computing $\{f_1, \dots, f_m\}$ in the same way that the action of S_n on the polynomial ring permutes $\{f_1, \dots, f_m\}$ (by induction on the structure of the circuit). Thus the span of f_1, \dots, f_m is an S_n -representation V .

Finally, we show that V is a direct sum of irreducible representations of shapes λ with $\lambda_1 \geq n-k$. Since C has support size k , the gate computing f_1 is fixed by a subgroup isomorphic to S_{n-k} ; thus the stabilizer H of f_1 contains S_{n-k} . Since $\{f_1, \dots, f_m\}$ spans V , by Obs. 2.4 there is an S_n -equivariant linear map $\varphi: \mathbb{F}[S_n/H] \rightarrow V$, such that the image of the trivial coset is H is f_1 . By equivariance, the S_n -orbit of f_1 is contained in the image of φ , and since the orbit of f_1 spans V , φ is surjective. Since $S_{n-k} \leq H$, there is an S_n -equivariant surjection $S_n/S_{n-k} \rightarrow S_n/H$, and which extends to a surjective S_n -equivariant linear map $\mathbb{F}[S_n/S_{n-k}] \rightarrow \mathbb{F}[S_n/H]$. Composing these, we get an S_n -equivariant linear surjection $\mathbb{F}[S_n/S_{n-k}] \rightarrow V$. By Ellis–Friedgut–Pilpel [EFP11, Thm. 7], $\mathbb{F}[S_n/S_{n-k}]$ only contains irreducible representations of shape λ with $\lambda_1 \geq n-k$. Thus the same is true for any quotient of $\mathbb{F}[S_n/S_{n-k}]$ as an S_n -representation, and in particular is true for V . This completes the proof. \square

5 Symmetric circuit size of separating modules and Weisfeiler–Leman

In this section we show that symmetric algebraic circuits of size $n^{\Theta(k)}$ and depth r are essentially equivalent to $\Theta(r)$ rounds of the $\Theta(k)$ -WL algorithm. We will also prove an analogous statement for Boolean threshold circuits—indeed, the statement for Boolean threshold circuits will be used in the course of the proof for algebraic circuits. In the Boolean setting, one direction of the equivalence is already known:

Theorem 5.1 (Dawar–Wiltschach, cf. [DW25, paragraph between Thms. 15 and 16] and Anderson–Dawar [AD17b, Lem. 12(2)]). *Let G, H be two graphs. If there is a rigid, square-symmetric Boolean (threshold) circuit C of fan-in at most s , support size at most $k \leq n/4$, and depth Δ such that $C(G) = 1$ and $C(H) = 0$, then G and H are separated by $O(\Delta(k + \log s / \log n))$ rounds of $O(k)$ -WL.*

Proof sketch. Essentially, this is already implicit in the proofs of Anderson–Dawar [AD17b, Sec. 4.4]. For the bound on WL dimension, this was pointed out explicitly by Dawar & Wiltschach [DW25, paragraph between Thms. 15 and 16]. The observation about depth & rounds is implicit in Anderson–Dawar but follows immediately from the proof. We give a brief sketch here.

They inductively construct FPC formulas for the constants $\{0, 1\}$, variables x_{ij} (these are “relational queries of arity 2” in their language, corresponding to the edge relation $E(i, j)$), OR gates,

AND gates, NAND gates, and Majority gates. The 0 and 1 formulas are quantifier free, the relational gates in our setting are all arity 2 since the only relation in the FO language of graphs is the edge relation, so in our setting their arity “ r ” is equal to 2. Examining the formulas on [AD17b, pp. 544–545], one sees that they each introduce no more than $O(r + k + \log_n s) = O(k + \log s / \log n)$ additional quantifiers. Thus, they translate threshold circuits of depth Δ to FPC formulas of depth $O((k + \log s / \log n)\Delta)$. Since quantifier depth and number of variables in FPC correspond to rounds and dimension in WL, respectively [CFI92], this completes the proof sketch. \square

5.1 Symmetric Algebraic Circuits Simulate WL

In this subsection we will prove the following theorem by taking a Boolean circuit which simulates WL and converting it to a rigid square-symmetric algebraic circuit that also simulates WL.

Theorem 5.2. *The r -round k -dim WL algorithm can be simulated by separating modules (in fact, invariant polynomials) computed by rigid square-symmetric algebraic circuits of size $rn^{O(k)}$ and depth $O(r)$.*

In the setting of Boolean threshold circuits, we will also use the following key result from Dawar & Wilsenach. The statement involves one new bit of terminology: when a circuit C is acted on by a group, we say the *orbit size* of C is the maximum size of any orbit of gates in C .

Theorem 5.3 (Dawar & Wilsenach [DW22, Thm. 4.10]). *Let C be a rigid square-symmetric (threshold) circuit of order $n > 8$. For every $k \leq n/4$, if the orbit size of C is less than $\binom{n}{k}$, then the support size of C is less than k .*

First, we look at a seemingly-stronger definition of rigidity from [AD17b] (see their Definition 5 and Proposition 1) that is easier to apply and turns out to be equivalent to the definition of rigidity from [DW25]. This definition uses the following notion of twins:

Definition 5.4 (Twins). Two gates $g, g' \in C$ are *in-twins* if they are the same type of gate and have the exact same set of inputs: for all $g'' \in C$, we have $(g'', g) \in C \iff (g'', g') \in C$. They are *twins* if they are in-twins and also out-twins: for all $g'' \in C$, we have $(g, g'') \in C \iff (g', g'') \in C$.

Proposition 5.5 (cf. [AD17b, Proposition 1]). *Let $G \leq S_n$ be a permutation group and C a Boolean or algebraic circuit on n variables that is G -symmetric (or even just a directed acyclic graph with n source nodes). If C has twins it is not G -rigid, and if C is not G -rigid then it has in-twins.*

Proof. First, suppose C has twin gates g, g' . Then there is an automorphism γ of C that swaps the two gates g, g' and fixes all the other gates. In particular, this means that both γ and the trivial automorphism are extensions of the trivial permutation of the inputs to the whole circuit, which is the identity element of G , so C is not G -rigid.

Conversely, suppose C is not G -rigid. Then there is some $\pi \in G$ and two distinct automorphisms α, β of C that both agree with π in how they act on the inputs. Then $\gamma := \alpha\beta^{-1}$ is a nontrivial automorphism of C that fixes all the inputs. Since γ is nontrivial, it must move at least one gate. Let g be a gate that is moved by γ such that none of the inputs to g are moved by γ ; such a g must exist by structural induction, because none of the inputs to C are moved by γ , yet γ moves at least one gate. Since γ is an automorphism of C that does not move the inputs to g , we have that $\gamma(g)$ and g are in-twins. \square

The Boolean circuit we start with is constructed by Grohe and Verbitsky. They construct an $r + 2$ layer Boolean threshold circuit in which the first layer simulates the initial coloring step of WL, the next r layers simulate each round of WL refinement, and the last layer computes the output. We want to prove that any permutation of inputs induces exactly one automorphism of the circuit. This will not quite be true of their circuit as written, but we will show how to modify their circuits to ensure this property.

Theorem 5.6 (Grohe–Verbitsky [GV06, Theorem 3.2]). *The r -round k -dim WL algorithm can be implemented by a logspace uniform family of Boolean threshold circuits of depth $O(r)$ and size $O(rn^{3k})$ where $k \geq 2$ and $r = r(n)$.*

Proof sketch. Layer 0 is a TC^0 circuit with $2k^2 \cdot 2n^k$ output gates which computes the initial coloring of tuples in both graphs based on marked automorphism types. Each marked automorphism type can be represented with a bit-string of length $2k^2$ and there are $2n^k$ tuples in both graphs. Then, each tuple is assigned a color with a name which is the index of the first tuple with the same color.

Layers 1 through r are each copies of a TC^0 circuit that does the WL iteration. It does this by checking two conditions. The first is that any pair of tuples with different colors, either in the same graph or one in either graph, must have different colors after passing through the layer. For any pair of tuples, this can be computed by a single AND gate with fan-in $2n^k$ for each color. Each AND gate of this type can run in parallel with constant depth. The second condition is that for any pair of tuples with one tuple in either graph, if the multiset of colors of i -neighboring tuples is distinct, then they get distinct colors after passing through the layer. This is done with three nested AND gates where the first runs over colors, the second runs over nodes in each tuple, and the third runs over vertices in each graph.

The final layer compares the counts of each color to see that they are equal in the two graphs, using threshold gates to check numerical equality and an AND gate over the different types of colors. \square

Unfortunately, the way that the authors index the colors in their construction of the circuit means that their specific construction may not be rigid. We will also need to modify their construction to produce circuits that look at the two graphs separately from one another, rather than together in a single circuit.

Given the graphs G, H first look at which k -marked isomorphism types actually occur in G , and number the initial colors of G lexicographically. Then build a circuit that is similar to the Grohe–Verbitsky circuit, except “tailored” to G , in that the initial colors aren’t determined the same way as the original construction, but rather we decide what the indexing of initial colors will be based on the k -marked isomorphism types we see in G . Call that circuit $C_{G,0}$.

Then we use that circuit $C_{G,0}$ on H as well, just to count how many of each marked isomorphism type occur by picking a marked isomorphism type where they differ and extending $C_{G,0}$ to output the count of that isomorphism type and subtracting off the count of that isomorphism type in G . If those counts are different, then we get a representation out of $C_{G,0}$ that vanishes on G but not H .

If those counts are the same, we do the same procedure again, but instead simulating round 1. We use the multi-output circuit $C_{G,0}$ as input to round 1, then look at the $\leq n^k$ distinct colors that occur in G and build a new circuit $C_{G,1}$ that has those particular colors “baked in” in lexicographic order. We then apply $C_{G,1}$ to H . If they differ, we pick a color where the counts differ and get a representation as above. If the Grohe–Verbitsky circuit distinguished G from H , then there is some round r such that $C_{G,r}$ distinguishes G from H .

Lemma 5.7. *The modified Grohe–Verbitsky circuit is rigid and square-symmetric.*

Proof. Importantly, the construction does not have any duplicate nodes in any layer. Every pair of nodes in one layer compute different things, and so their inputs are different. Thus, this circuit has no in-twins and is therefore rigid (Proposition 5.5).

The circuit is square-symmetric because any permutation on inputs is in essence a permutation of labelings of vertices or vertices within a k -tuple. In either case, WL will output the same result, so there must exist a corresponding automorphism of the circuit. \square

Lemma 5.8. *Any Boolean threshold circuit C which is rigid square-symmetric under the action of any group G can be converted to an algebraic circuit which computes the same output of C and is rigid square-symmetric under the action of G .*

Proof. The construction uses the standard folklore conversion of replacing AND gates with multiplication gates, NOT gates with $1 - x$ gates, and threshold gates with appropriate interpolation gadgets (take the sum of the inputs, then apply Lagrange interpolation to get an algebraic circuit that outputs 1 if that sum is greater than the threshold and 0 if it is less). Replacing gates by gadgets in this way preserves rigidity and square-symmetry between structures in the algebraic circuit. Within the gadgets there are no twins and permuting inputs will still correspond to at least one automorphism since the structure is otherwise preserved. \square

Proof of Thm. 5.2. The modified Grohe–Verbitsky circuit still computes the same output as the original Grohe–Verbitsky circuit, so by Lem. 5.7 the modified Grohe–Verbitsky circuit is a rigid square-symmetric circuit, and furthermore by Thm. 5.6 it also simulates WL. We can convert the modified Grohe–Verbitsky circuit into a rigid square-symmetric algebraic circuit C^* which computes the same output by Lem. 5.8. C^* is a single-output rigid square-symmetric circuit, so in particular it computes an invariant polynomial. This completes the proof. \square

5.2 WL Simulates Symmetric Circuits

To prove the opposite direction we will use a result of Dawar & Wilsenach on simulating algebraic circuits by Boolean threshold circuits. We note that in the following result, there is no bound assumed on the degree of the algebraic circuit, and no bound claimed on the size of the resulting Boolean circuit. Rather, the bounds on the resulting Boolean circuit are only on its orbit size, and (we observe) on its depth. We have stated their result in notation consistent with ours.

Theorem 5.9 (Dawar & Wilsenach [DW25, Theorem 11]). *Let G be a group acting on a set of variables X . If C is a G -symmetric constant-free algebraic circuit over a field \mathbb{F} with variables X and $B \subseteq \mathbb{F}$ is a finite non-empty set, then there is a G -symmetric Boolean threshold circuit C_{bool} on variables X such that*

1. For all $\{0, 1\}$ assignments $\nu: X \rightarrow \{0, 1\}$, we have $C(\nu(X)) \in B \iff C_{bool}(\nu(X)) = 1$;
2. C and C_{bool} have the same largest G -orbit size;
3. C and C_{bool} have the same depth.

The statement about depth is not contained in their statement, but follows directly from their proof. Their statement was also only for single-output circuits, but it is immediate from their proof that the same construction with the above properties works for multi-output G -symmetric circuits.

This lets us prove:

Theorem 5.10. *For each $1/\log_2 n < \varepsilon < 1$ and $k \leq \varepsilon n^{1-\varepsilon}$, if two n -vertex graphs G, H admit a separating module with symmetric algebraic circuits of bit-size $n^{O(k)}$, then they can be distinguished by $O(k/\varepsilon)$ -WL.*

Proof. Let C be a (square-)symmetric algebraic circuit of bit-size $n^{O(k)}$ computing a spanning set for a separating module V . Since C is rigid square-symmetric of size $n^{O(k)}$, the maximum size of every orbit is at most $n^{O(k)}$.

First we convert C into a *constant-free* algebraic circuit of size $n^{O(k)}$, that is still rigid and (square-)symmetric. For this, we observe that the construction in Lemma 2.3 uses gadgets that are internally rigid. If any of the gates within those gadgets are in-twins, we may merge those constant gates into single constants. This merging process has the effect that the gates that syntactically compute constants (meaning the entire sub-DAG rooted at that gate only has constants on all of its leaves) all compute distinct constants, and thus cannot be moved by any automorphism of the circuit. The size of the resulting circuit is $n^{O(k)} \cdot O(k \log n)$, and since $k \leq n$, the latter is $n^{O(k+2)} = n^{O(k)}$.

Now let C_{bool} be the corresponding Boolean circuit guaranteed by Theorem 5.9 with the finite set B being the singleton $\{0\}$ (since what we care about is whether our polynomials evaluate to 0 or not). By the properties there, C_{bool} also has maximum orbit size $n^{O(k)}$ (even though its total circuit size can be much larger). Thus, for $k < \varepsilon n^{1-\varepsilon}$, C_{bool} has support size less than $O(k/\varepsilon)$, by Lemma 2.1.

By Theorem 5.1 a Boolean threshold circuit of support size $O(k/\varepsilon)$ translates to a formula with $O(k/\varepsilon)$ variables and so can be distinguished by $O(k/\varepsilon)$ -dimensional WL. \square

Combining the preceding theorem with Thm. 5.2, we get:

Corollary 5.11. *For any $1/\log_2 n < \varepsilon < 1$ and $k \leq \varepsilon n^{1-\varepsilon}$, k -WL is simulated by separating modules of symmetric algebraic circuits of bit-size $n^{O(k)}$, which in turn are simulated by $O(k/\varepsilon)$ -WL. In particular, when ε is constant, then $\Theta(k)$ -WL is equivalent in distinguishing power to separating modules of symmetric algebraic circuits of bit-size $n^{\Theta(k)}$.*

The same results hold if everywhere we replace “separating module” by “separating invariant.”

One advantage of Dawar & Wilsenach’s construction (reproduced as Theorem 5.9 above) is that it produces circuits—over arbitrary fields—with well-controlled orbit size, even if the overall size of the resulting circuit could be exponential. However, in the context of the integers, and if the algebraic circuit additionally has polynomial degree, we give an alternative construction that additionally maintains polynomial size.

For this result, we come to a distinction that has not yet been relevant in the rest of the paper: algebraic circuits compute polynomials in $\mathbb{F}[X]$, even though when applied to adjacency matrices we only “care” about the resulting function in $\mathbb{F}[X]/\langle x_{ij}^2 - x_{ij} \rangle$. In the latter ring, degree is at most $O(n^2)$, while in $\mathbb{F}[X]$ degree can be unbounded. However, if a symmetric algebraic circuit computes a spanning set of a test module $V \leq \mathbb{F}[X]$ that is not identically zero on all adjacency matrices, then $V \pmod{\langle x_{ij}^2 - x_{ij} \rangle}$ is a nonzero test module in $\mathbb{F}[X]/\langle x_{ij}^2 - x_{ij} \rangle$ (Lemma 2.8).

Our next result works for degrees up to $n^{O(k)}$, where degree is measured in $\mathbb{F}[X]$, as above (note: much too large for Theorem 3.11 to be relevant). In that case, we can use a more direct folklore simulation of algebraic circuits over the integers by Boolean threshold circuits, as in the following lemma.

Lemma 5.12. *Suppose C is a rigid, square-symmetric, constant-free, multi-output algebraic circuit of size S and depth Δ over the integers, such that the polynomial computed at each gate has degree at most D . Then there is a Boolean threshold circuit of total size $O(DS^2)$ and depth $O(\Delta)$ whose multiple output bits, when interpreted as the binary expansion of an integer, computes the same values as C on all $\{0, 1\}$ inputs, and furthermore such that the resulting Boolean circuit is also rigid square-symmetric.*

Proof. Since the total size of C is S and the degree of the polynomial at each gate is at most D , the values of integers computed at each gate on any $\{0, 1\}$ -inputs have at most $O(D)$ bits. Thus we can simulate C on $\{0, 1\}$ inputs by a Boolean threshold circuit C' of size $O(S^2D)$ and depth $O(\Delta)$: each multiplication gate and each addition gate can be simulated by TC^0 gadgets.

To see that the resulting circuit is still rigid and square-symmetric, we briefly recall the TC^0 gadget for iterated addition; we follow the exposition from [Kay15]. Let a_1, \dots, a_m be integers each with m bits (we use m so it doesn’t conflict with n elsewhere in the proof), let s refer to their sum, and let c_0, \dots, c_m be the carry integers. Recall that addition of binary numbers is done by computing the carries and then computing a large parity for each bit of the sum. Since each c_i is an integer, only their least significant bit will affect $s[i]$. Thus, $s[i] = \text{PARITY}(c_i[0], a_1[i], \dots, a_m[i])$. This computation of $m + 1$ bits for the parity can be done in constant depth using threshold gates since $\text{PARITY} \in \text{TC}^0$. Now, we just need to show that computation of carries is doable in constant depth using threshold gates. Consider a carry c_j and note that it only depends on the j least significant bits of each of the a_i ’s. Let the sum of these bits be t with length k . Since each term of the sum is

integer smaller than 2^j (j bits) and there are m of them for each a_i , the total number of bits of t is $\log_2(m2^j) = \log m + j$. Then, we get c_j by dropping the j least significant bits of t , so c_j has $\log n$ bits and k is fixed for a fixed m . Note that

$$c_j[k] = 1 \iff (2^{k+1} \geq t \geq 2^k)$$

$$c_j[k-1] = 1 \iff (2^k - 1 \geq t \geq 2^{k-1}) \vee (2^k + 2^k - 1 \geq t \geq 2^k + 2k - 1)$$

and so on. Since we only care about $c_j[0]$, which requires $2 \cdot 2^{k-j} \leq 2 \cdot 2^{\log m} = O(m)$ bits, we can compute $c_j[0]$ using $O(m)$ threshold gates and depth 2. Then, computing the parity can be done in $O(m^2)$ gates and depth 3. Thus the whole gadget has size $O(m^3)$ and depth 5.

Recall that C is rigid square-symmetric. To show that C' is also rigid square-symmetric, we show that converting addition gates to iterated addition gates maintains rigidity and symmetry. (The proof for multiplication gates is similar.) For this, we want to show that the only way to swap individual gates in each gadget is to swap the whole gadget, and that there is no automorphism of the circuit that swaps two gates inside of a single gadget. The gates computing $c_i[0]$ are all quite different from one another, so there are many direct ways to show they cannot be moved among one another by any automorphism, for example: they are in different layers, have different in-degrees, etc. Furthermore, because each bit $s[i]$ of the sum depends on $c_i[0]$, no automorphism can move any $s[i]$ gates. Thus, C' is rigid. Finally, suppose an automorphism permutes the order of the integers that are the inputs to C' . This corresponds to exactly one isomorphism of C' , which permutes the entire gadgets in the corresponding order. Thus, C' is square-symmetric. The depth of this circuit is a constant (from TC^0 gadgets) times Δ . \square

Theorem 5.13. *For any $1/\log_2 n < \varepsilon < 1$ and $k \leq \varepsilon n^{1-\varepsilon}$, if G, H are two n -vertex graphs distinguished by a separating module computed by a constant-free symmetric algebraic circuit of size $n^{O(k)}$ and depth Δ , in which every gate has degree $n^{O(k)}$, then G and H are distinguished by $O(\Delta)$ rounds of $O(k/\varepsilon)$ -WL.*

Proof. Let C be the symmetric algebraic circuit as in the statement of the theorem. By Lemma 5.12, C is faithfully simulated by a Boolean threshold circuit C' of size $n^{O(k)}$ and depth $O(\Delta)$ that is rigid and square-symmetric. Furthermore, suppose without loss of generality that C vanishes on G but not on H ; then the same is true of C' . Now we add one new OR gate to the top of C' , connected to all all of the outputs of C' ; call the resulting circuit \widehat{C} . Then $\widehat{C}(H) = 1$ and $\widehat{C}(G) = 0$. Furthermore, \widehat{C} is still a rigid square-symmetric circuit of size $n^{O(k)}$ and depth $O(\Delta)$, but now has just one output. By Lemma 2.1, this means that \widehat{C} has size, and hence orbit size, at most $n^k < \binom{n}{k/\varepsilon}$ and so by Thm. 5.3, \widehat{C} has support size $O(d/\varepsilon)$. Thus, by Thm. 5.1 we get that G and H can be separated by $O(\Delta)$ rounds of $O(k/\varepsilon)$ -WL. \square

6 Multiplicity obstructions for graph non-isomorphism

6.1 Additional preliminaries for this section

Group actions. For a group Γ , a Γ -set is a set X together with an action of Γ (equivalently, a group homomorphism from Γ to the group of permutations of X). Two Γ -sets X, Y are said to be isomorphic if there is a bijection $f: X \rightarrow Y$ such that $f(\gamma x) = \gamma \cdot (f(x))$ for all $x \in X$, where on the left-hand side γx is the action of Γ on X , and on the right-hand side $\gamma \cdot (f(x))$ is the action of Γ on Y . X is said to be a *transitive* Γ -set if for all $x, x' \in X$, there exists $\gamma \in \Gamma$ such that $\gamma \cdot x = x'$.

With one exception towards the end of this background section, that we will clearly mark, all of the material here is completely standard; see, for example, any standard textbook on group or representation theory, e. g., [FH91, DF04].

For representations of finite groups Γ over the complex numbers, the isomorphism type of the representation $\rho: \Gamma \rightarrow GL_d(\mathbb{C})$, where d is the dimension, is determined by its *character*, which by definition the function

$$\chi_\rho: \Gamma \rightarrow \mathbb{C} \quad \chi_\rho(g) := \text{tr}(\rho(g)).$$

An immediate consequence of this definition is that because the trace of a matrix remains the same after conjugation, if $g, h \in \Gamma$ are in the same conjugacy class, $\chi_\rho(g) = \chi_\rho(h)$. Thus, when $\Gamma = S_n$, $\chi_\rho(g)$ depends only on the cycle type of g since the conjugacy classes in S_n are exactly determined by cycle type.

Because there is a bijection between characters and representations, just as we speak of the multiplicity of an irrep in another representation, if χ is the character of an irrep V and χ' is the character of another representation W , by the multiplicity of χ in χ' we mean the multiplicity of V in W . Similarly, we speak of the character χ itself being irreducible when it is the character of an irrep. The character of most interest to us in this section is the character of the trivial representation, which we will denote χ_1 . The trivial representation $\mathbf{1}: \Gamma \rightarrow GL_1(\mathbb{C})$ sends all group elements to the 1-dimensional identity matrix. Thus, $\chi_1(g) = 1$ for all $g \in \Gamma$. The trivial representation is an irrep.

Character table. It is a standard theorem that the number of irreps up to isomorphism is equal to the number of conjugacy classes. This lets us arrange all the characters of all irreps into a square table with side length equal to the number of conjugacy classes of Γ , called the *character table*. We index its columns by conjugacy classes and index its rows by the irreps of Γ . For a column corresponding to the conjugacy class of $g \in \Gamma$ and a row corresponding to the irrep V , we put the character value $\chi_V(g)$ in the $(V, [g])$ entry of the table (recall that this value is the same for any choice of element in the conjugacy class $[g]$).

Theorem 6.1. *The character table is always invertible for any Γ .*

Inner products of characters. The characters for any representation have a natural notion of an inner product

$$\langle \chi, \chi' \rangle_\Gamma = \frac{1}{|\Gamma|} \sum_{g \in \Gamma} \chi(g) \overline{\chi'(g)}$$

where the bar denotes complex conjugation. Recall that the character values for S_n are all integer-valued (see Fact 2.5), so complex conjugation does not affect the inner product in our setting. If χ is the character of an irrep, then the inner product $\langle \chi, \chi' \rangle$ is equal to the multiplicity of χ in χ' .

Representations of subgroups. For any group Γ with subgroup Σ , we can restrict any character of Γ down to any character of Σ , denoted $\text{Res}_\Sigma^\Gamma \chi_\rho$ where ρ is an irrep of Γ , and we can induce any character of Σ up to any character of Γ , denoted $\text{Ind}_\Sigma^\Gamma \chi_{\rho'}$ where ρ' is an irrep of Σ .

Theorem 6.2 (Frobenius Reciprocity). $\langle \text{Ind}_\Sigma^\Gamma \chi_{\rho'}, \chi_\rho \rangle_\Gamma = \langle \chi_{\rho'}, \text{Res}_\Sigma^\Gamma \chi_\rho \rangle_\Sigma$

The one straightforward fact that we will use that might not be covered in standard textbooks (but is not beyond the difficulty of an exercise) is the following. The coset representation $\mathbb{F}[S_n/\Gamma]$ is equal to the induced representation in S_n of the trivial representation in Γ , i. e. $\mathbb{F}[S_n/\Gamma] \cong \text{Ind}_\Gamma^{S_n} \mathbf{1}$.

6.2 Computing multiplicities

We begin this section by showing that the representation-theoretic multiplicities supported on a graph's orbit, up to degree d , can be computed in $n^{O(d)}$ time. (Or up to support-degree d in $2^{O(d^2)} n^{O(d)}$ time.)

Theorem 6.3. *Let \mathbb{F} be a field of characteristic 0 or $> n$. There is an algorithm which, given a graph G and a positive integer d , computes the multiplicities of every S_n -irrep that occurs in $\mathbb{F}[S_n \cdot G]$ up to degree d in time $n^{O(d)}$.*

In particular, there is an algorithm that, given two n -vertex graphs G, H and an integer d as input, will find a multiplicity obstruction in degree $\leq d$ if one exists, or report that none does, in time $n^{O(d)}$.

As with Lemma 3.8, an analogous result holds with degree replaced by support-degree, but with runtime $2^{O(d^2)}n^{O(d)}$ (the number of monomials of support-degree at most d).

Proof. The third paragraph of the proof of Theorem 3.1 already says directly that we can compute all irreps in $\mathbb{F}[X]$ up to degree d that vanish on G (resp., H), as well as which ones are equivalent to which other ones as abstract representations. We can then apply the algorithm of Brooksbank & Luks [BL08] again to tell which irreps that vanish on G are equivalent to which ones that vanish on H . At that point, the algorithm can directly check if any of the multiplicities differ. \square

6.3 Characterizing multiplicities

Definition 6.4 (Cycle index). The *cycle index* of a permutation group $G \leq S_n$ is the multivariate generating function for the multiset of its cycle types. That is, if $c_i(\pi)$ denotes the number of i -cycles in $\pi \in S_n$, the cycle index of G is

$$cyc_G(x_1, \dots, x_n) := \sum_{\pi \in G} x_1^{c_1(\pi)} x_2^{c_2(\pi)} \dots x_n^{c_n(\pi)}.$$

We will not really use the generating function aspect of the cycle index; rather, we use it as a convenient container for the multi-set of cycle-types of a permutation group (and we use the name to help connect it to literature on the cycle index).

Theorem 6.5. For all $G \leq S_n$, the collection of all multiplicities of S_n -representations in the permutation representation S_n/G determine and are determined by the cycle index of G .

We first prove an immediate corollary about the power of multiplicities to distinguish non-isomorphic graphs:

Corollary 6.6. Two n -vertex graphs G and H are separated by a multiplicity obstruction if and only if the cycle indices of $\text{Aut}(G) \leq S_n$ and $\text{Aut}(H) \leq S_n$ are distinct.

Proof. Let $R = \mathbb{F}[x_{ij}]/\langle x_{ij}^2 - x_{ij} \rangle$ and let $I_G \subseteq R$ be the vanishing ideal of the orbit $S_n \cdot G$ (and similarly for I_H). Then for any irrep χ , we have $\text{mult}_\chi(\mathbb{F}[S_n \cdot G]) = \text{mult}_\chi(R) - \text{mult}_\chi(I_G)$, so the multiplicities of irreps that vanish on $S_n \cdot G$ give equivalent information to the multiplicities that occur in the coordinate ring of the orbit, $\mathbb{F}[S_n \cdot G]$. Now, because the orbit is a finite set, and the Orbit–Stabilizer Theorem gives an S_n -equivariant isomorphism $S_n \cdot G \rightarrow S_n/\text{Aut}(G)$ (the latter being the coset space), as S_n representations we have $\mathbb{F}[S_n \cdot G]$ and $\mathbb{F}[S_n/\text{Aut}(G)]$ are equivalent: $\text{mult}_\chi \mathbb{F}[S_n \cdot G] = \text{mult}_\chi \mathbb{F}[S_n/\text{Aut}(G)]$. By Theorem 6.5, the latter multiplicities contain the same information as the cycle index of $\text{Aut}(G)$. Thus G and H are distinguished by S_n -multiplicities in their vanishing ideals I_G, I_H if and only if $\text{Aut}(G)$ and $\text{Aut}(H)$ have distinct cycle indices. \square

Proof of Theorem 6.5. Let $G, H \leq S_n$ be two subgroups of S_n . Let u be the column vector indexed by cycle types, where for each cycle type c , u_c is the number of elements of G of cycle type c . We will refer to u as the “cycle index vector” of G . Similarly, let v be the cycle index vector of H . By construction, the cycle index vector and the cycle index polynomial “contain the same data” (just represented by different types of objects).

We first show that if either the multiplicities or the cycle indices are the same, then $|G| = |H|$. Note that $\sum_c u_c = |G|$, so if the cycle indices are the same, then $|G| = \sum_c u_c = \sum_c v_c = |H|$. On the other hand, if the multiplicities of S_n -irreps in $\mathbb{F}[S_n/G]$ are the same as those in $\mathbb{F}[S_n/H]$, then we have $|S_n/G| = \dim \mathbb{F}[S_n/G] = \dim \mathbb{F}[S_n/H] = |S_n/H|$, and therefore $|G| = |H|$.

Now we show how to recover the multiplicities from the cycle index vectors (and then at the end, because this uses the character table and the character table is invertible, we will get the other direction almost for free). Let χ be an irreducible character of S_n for which we wish to know its multiplicity in $\mathbb{F}[S_n/G]$. As a representation, the latter is the induced representation $\text{Ind}_G^{S_n} \chi_1$, where χ_1 is the trivial character of G . So we seek the inner product $\langle \chi, \text{Ind}_G^{S_n} \chi_1 \rangle_{S_n}$, where we have used the subscript S_n to remind the reader that here this is the inner product of S_n -characters. By Frobenius Reciprocity (Theorem 6.2), we have

$$\text{mult}_\chi(\mathbb{F}[S_n/G]) = \langle \chi, \text{Ind}_G^{S_n} \chi_1 \rangle_{S_n} = \langle \text{Res}_G^{S_n} \chi, \chi_1 \rangle_G.$$

To calculate the latter, note that χ_1 on G is just the constant-1 function, and $\text{Res}_G^{S_n}(\chi)(g) = \chi(g)$, which is completely determined by the cycle type of G . Thus we seek to compute

$$\langle \text{Res}_G^{S_n} \chi, \chi_1 \rangle_G = \frac{1}{|G|} \sum_{g \in G} \chi(g).$$

Since $\chi(g)$ only depends on the cycle type of g , the latter sum is a scaled version of summing the χ -th row of the character table M of S_n , with each entry weighted by the number of elements of G with that cycle type. Thus we have that the χ -th entry of Mu (where u is the cycle index vector of G) is the multiplicity of χ in $\mathbb{F}[S_n/G]$:

$$\text{mult}_\chi(\mathbb{F}[S_n/G]) = \frac{1}{|G|} (Mu)_\chi$$

Putting these together, we get that the vector $\frac{1}{|G|}Mu$ is the vector of all the multiplicities of S_n irreps in $\mathbb{F}[S_n/G]$. Thus, if G and H have the same cycle index, then $u = v$ and (by the first paragraph above) $|H| = |G|$, so the vector of multiplicities $(1/|G|)Mu$ for G is equal to $(1/|H|)Mv$ for H .

Conversely, if the multiplicity vectors are equal, then we have $(1/|G|)Mu = (1/|H|)Mv$. As $|G| = |H|$ (again by the first paragraph above), and M is invertible (Theorem 6.1), we get that $u = v$, and thus G and H have the same cycle index. \square

6.4 Limitations of multiplicity obstructions

Observation 6.7. Almost all pairs of graphs are not distinguishable by multiplicities but are distinguishable by separating modules. The same is true for almost all pairs of random regular graphs of degree $3 \leq d \leq n - 4$.

Proof. Since all orbits of S_n are finite, any two distinct orbits (=non-isomorphic graphs) are in fact separated by invariants (e. g., [DK15, Thm. 2.3.6]), which is a particular case of separating modules. On the other hand, almost all graphs have trivial automorphism group [ER63], so by Cor. 6.6 any two such graphs are not distinguished from one another by multiplicity information alone. For random regular graphs, we use the analogous result on automorphism groups in that setting [KSV02]. \square

As pointed out in the introduction, it is thus natural to ask whether all graphs that are characterized by their symmetries are distinguishable by multiplicities. Here we show that in constant degree, only finitely many graphs can be distinguished by multiplicities:

Proposition 6.8. *For any fixed degree d , as $n \rightarrow \infty$, the multiplicities of irreps in degree $\leq d$ can distinguish at most finitely many pairs of non-isomorphic graphs. In particular, there is no constant d such that multiplicity information in degree $\leq d$ distinguishes all pairs of non-isomorphic graphs that are characterized by their symmetries.*

Proof. The polynomials of degree d in the polynomial ring $\mathbb{F}[x_{ij} : i, j \in [n]]$ are, as a representation of S_n , $\text{Sym}^d(V_n \otimes V_n)$, where V is the defining n -dimensional representation of S_n . By [CEF15, Prop 3.4.1], the sequence V_n is the sequence of representations of a finitely generated FI-module. Then by [CEF15, Thm. 3.4.2], the same is true of $V_n \otimes V_n$, and then by [CEF15, Prop. 3.4.3] the same is true of $\text{Sym}^d(V_n \otimes V_n)$ for any fixed value of d . By [CEF15, Thm. 1.13], this sequence of representations is thus “uniformly representation stable” in the sense of Church & Farb [CF13]. Part of the definition of the latter is that there is a finite list of partitions $\lambda_1, \dots, \lambda_k$ and multiplicities $m_1, \dots, m_k \in \mathbb{N}$ —all of which depend only on d , not on n —such that for all sufficiently large n , $\text{Sym}^d(V_n \otimes V_n) = \bigoplus_{i=1}^k S_{\lambda_i[n]}^{\oplus m_i}$, where $\lambda[n]$ is the extension of λ to a partition of n by increasing the first part of the partition so the whole partition adds up to n . In particular, the total multiplicity of all irreps in $\text{Sym}^d(V_n \otimes V_n)$ is $\sum_{i=1}^k m_i$, which is a constant that depends only on d , not on n . Thus the total number of possibilities for the multiplicities of each irrep in degree d is $\prod_{i=1}^k (m_i + 1)$, which is again a constant that depends on d but not on n , so the multiplicities in degree d (and hence, *a fortiori*, in degree $\leq d$) can partition all isomorphism classes of graphs into only $f(d)$ sub-classes, where f is a function that depends only on d , not on n .

Those multiplicities are an upper bound on the multiplicities in the quotient ring $\mathbb{F}[x_{ij}]/\langle x_{ij}^2 - x_{ij} \rangle$, so we similarly get a bound there that depends only on d and not on n . (We remark that the family of ideals $\langle x_{ij}^2 - x_{ij} : i, j \in [n] \rangle$ is a finitely generated FI-ideal, so the above argument also applies directly to the quotiented ring as well.)

For the “in particular,” we recall that a disjoint union of k many ℓ -cliques is characterized by its symmetries (for any $k, \ell \geq 1$). For any factorization $n = k\ell$ we get such a graph, and graphs with different pairs (k, ℓ) are non-isomorphic (and not isomorphic to one another’s complements either). For example, for n a product of μ distinct primes, there are 2^μ different pairs (k, ℓ) with $k\ell = n$, hence 2^μ mutually non-isomorphic n -vertex graphs that are characterized by their symmetries. Choosing $\mu > \log_2 f(d)$ we find that these graphs that are characterized by their symmetries cannot all be distinguished from one another by multiplicities up to degree d . \square

6.5 Multiplicity obstructions are stronger than occurrence obstructions

By analogy with occurrence obstructions in GCT, given two n -vertex graphs G, H , we say that an irrep V of S_n is an *occurrence obstruction* if

$$\text{mult}_V(\mathbb{F}[S_n/\text{Aut}(G)]) = 0 \neq \text{mult}_V(\mathbb{F}[S_n/\text{Aut}(H)])$$

(or vice versa). (The reason for the name is that the occurrence of V in the coordinate ring of the orbit of H obstructs an isomorphism with G .) By the “occurrence information” for a group $\Gamma \leq S_n$ we mean the set (not multiset) of which equivalence classes of irreps occur in $\mathbb{F}[S_n/\Gamma]$ with non-zero multiplicity.

Lemma 6.9. *Suppose $\Gamma_1, \Gamma_2 \leq S_k$; for $i = 1, 2$, let $\Gamma_i^{[n]}$ be the group $S_{n-k} \times \Gamma_i \leq S_n$. Then*

1. $\Gamma_1, \Gamma_2 \leq S_k$ have the same multiplicity information if and only if $\Gamma_1^{[n]}, \Gamma_2^{[n]} \leq S_n$ have the same multiplicity information; and
2. If $\Gamma_1, \Gamma_2 \leq S_k$ have the same occurrence information, then $\Gamma_1^{[n]}, \Gamma_2^{[n]} \leq S_n$ have the same occurrence information.

Proof. 1. We use the characterization in terms of cycle indices (Theorem 6.5). The cycle indices of Γ_i determine and are determined by those of $\Gamma_i^{[n]}$. More specifically, when viewed as multivariate generating functions, the cycle index of $G \times H$ is the product of the cycle indices of G and H . Writing $c(G)$ for the cycle index polynomial, we then have $c(\Gamma_i^{[n]}) = c(\Gamma_i)c(S_{n-k})$, from which it follows immediately that $c(\Gamma_1) = c(\Gamma_2) \iff c(\Gamma_1^{[n]}) = c(\Gamma_2^{[n]})$.

2. By Frobenius reciprocity, an irrep of S_n contains a Γ -invariant if and only if it occurs with non-zero multiplicity in $\mathbb{F}[S_n/\Gamma]$. We will show that each irrep of S_n contains a $\Gamma_1^{[n]}$ invariant if and only if it contains a $\Gamma_2^{[n]}$ invariant. Toward that end, let V be an irrep of S_n . Suppose the restriction of V to $\Gamma_1^{[n]}$ contains an invariant. Since $\Gamma_i^{[n]} = S_{n-k} \times \Gamma_i \leq S_{n-k} \times S_k$, we will first examine the restriction of V to $S_{n-k} \times S_k$. Suppose $\text{Res}_{S_{n-k} \times S_k}^{S_n} V = \bigoplus_i U_i \boxtimes V_i$, where each U_i is an irrep of S_{n-k} and each V_i is an irrep of S_k (recall that irreps of a direct product are tensor products of irreps of the factors, see Section 2.2). Then

$$\begin{aligned}
\text{Res}_{\Gamma_i^{[n]}}^{S_n} V &= \text{Res}_{\Gamma_i^{[n]}}^{S_{n-k} \times S_k} \left(\text{Res}_{S_{n-k} \times S_k}^{S_n} V \right) \\
&= \text{Res}_{\Gamma_i^{[n]}}^{S_{n-k} \times S_k} \left(\bigoplus_j U_j \boxtimes V_j \right) \\
&= \bigoplus_j \text{Res}_{\Gamma_i^{[n]}}^{S_{n-k} \times S_k} (U_j \boxtimes V_j) \\
&= \bigoplus_j \text{Res}_{S_{n-k}}^{S_{n-k}} U_j \boxtimes \text{Res}_{\Gamma_i}^{S_k} V_j \\
&= \bigoplus_j U_j \boxtimes \text{Res}_{\Gamma_i}^{S_k} V_j,
\end{aligned}$$

where the second to last line follows from the basic fact about restrictions of direct products (see Section 2.2).

Thus, the trivial $\Gamma_i^{[n]}$ representation occurs in $\text{Res}_{\Gamma_i^{[n]}}^{S_n} V$ if and only if there is some j such that

- (1) U_j is the trivial rep of S_{n-k} and (2) $\text{Res}_{\Gamma_i}^{S_k} V_j$ contains the trivial Γ_i representation.

But now, by our assumption that the occurrence information for $\Gamma_1, \Gamma_2 \leq S_k$ are the same, we have that $\text{Res}_{\Gamma_1}^{S_k} V_j$ contains a trivial representation if and only if $\text{Res}_{\Gamma_2}^{S_k} V_j$ does. Combining with the preceding paragraph, that gives us that V contains a $\Gamma_1^{[n]}$ invariant if and only if it contains a $\Gamma_2^{[n]}$ invariant. \square

Although the converse of Lemma 6.9(2) may well be true, we do not see how to prove it along the above lines. (The issue is that for each S_n -irrep V with a $\Gamma_i^{[n]}$ -invariant, all we conclude from assuming that $\Gamma_1^{[n]}$ and $\Gamma_2^{[n]}$ have the same occurrence information is that there exists a j such that V_j contains a Γ_i -invariant, but it is possible that this occurs in different V_j for $i = 1, 2$.)

Corollary 6.10. *If (G, H) is a pair of k -vertex graphs that are separated by multiplicities but not occurrence obstructions, then $G \sqcup K_{n-k}$ and $H \sqcup K_{n-k}$ are n -vertex graphs that are separated by multiplicities but not occurrence obstructions, for all $n > 2k$.*

Proof. If $n > 2k$, then $\text{Aut}(G \sqcup K_{n-k}) = \text{Aut}(G) \times S_{n-k}$, since G is too small to contain a clique of size $n - k$. The result then follows immediately from Lemma 6.9. \square

Thus finding infinitely many such examples reduces to finding a single example. For the sake of interest, we give a different (albeit related) construction of an infinite family of pairs of graphs that are separated by multiplicities but not occurrence obstructions. Our construction in the next proposition also has the advantage that it does not have the ‘‘gap’’ between k and $2k$ that would arise by applying the preceding corollary.

Proposition 6.11. *For all $n \geq 6$, there are pairs of simple undirected graphs (G_n, H_n) on n vertices such that G_n and H_n are separated by multiplicity obstructions but not by occurrence obstructions.*

For simple undirected graphs, the value of 6 is optimal: by explicit calculation, we found that there are no pairs of simple undirected graphs on $n \leq 5$ vertices that are separated by multiplicities but not separated by occurrences.

Proof. We exhibit one infinite family of pairs of graphs. Let $G_n = P_4 + [n - 4]$ be the disjoint union of a 4-vertex path (length 3) and $n - 4$ isolated vertices. We define H_n to be a large “dandelion”: a path on 4 vertices, with one endpoint connected to $n - 4$ vertices of degree 1 (which we refer to as the “seeds” of the dandelion). One may think of G_n as being H_n after the seeds of the dandelion were blown off.

Then we have $\text{Aut}(G_n) = C_2 \times S_{n-4}$, where C_2 acts on the 4-vertex path by reversing it, which has cycle type $(\bullet\bullet)(\bullet\bullet)$. For concreteness, we suppose the path to be on vertices $n - 3, n - 2, n - 1, n$, in order.

Similarly, for $n \geq 6$, we have $\text{Aut}(H_n) = S_{n-4}$, permuting the seeds of the dandelion arbitrarily. (At $n = 4$ we have $G_4 = H_4 = P_4$ and at $n = 5$, $H_n = P_5$.) For ease of comparison with $\text{Aut}(G_n)$, let us suppose the 4 non-seed vertices are $n - 3, n - 2, n - 1, n$, so $\text{Aut}(H_n)$ is the copy of S_{n-4} acting on $[n - 4]$.

Since the cycle indices of these groups are distinct (which follows from the groups having different sizes), by Theorem 6.5 we have that G_n and H_n are distinguished by a multiplicity obstruction.

It remains to show that they are not distinguished by occurrence obstructions. By standard representation theory (which we have recalled in the proof of Theorem 6.5) we have that an irrep V of S_n occurs in the permutation representation S_n/Γ if and only if V , when restricted to Γ , contains a non-zero Γ -invariant vector. To show that the occurrence information for $\text{Aut}(G_n)$ and $\text{Aut}(H_n)$ are the same, it is necessary and sufficient to show that for every irrep V_λ of S_n , V_λ contains $\text{Aut}(G_n)$ -invariants if and only if it contains $\text{Aut}(H_n)$ invariants. Since in our case $\text{Aut}(H_n) \leq \text{Aut}(G_n)$, for each S_n -irrep that contains an S_{n-4} invariant, we merely need to check that it also contains, among the S_{n-4} -invariants, an invariant for the additional copy of C_2 in $\text{Aut}(G_n)$, generated by $\sigma := (n - 3, n)(n - 2, n - 1)$.

Now, if $v \in V_\lambda$ is any vector, then $v + \sigma(v)$ is C_2 -invariant, and is non-zero if and only if $\sigma(v) \neq -v$. Furthermore, since the action of $C_2 = \langle \sigma \rangle$ commutes with the action of S_{n-4} , if v is S_{n-4} -invariant, then $v + \sigma(v)$ is also S_{n-4} -invariant (hence, $S_{n-4} \times C_2$ -invariant). Thus, it suffices to find an S_{n-4} -invariant vector v in an S_n irrep, and then show that $\sigma(v) \neq -v$.

From the branching rule for restriction from S_n down to S_{n-1} [Sag01, §2.8], and then iterated down to S_{n-4} , an S_n irrep contains an S_{n-4} invariant if and only if the corresponding partition λ has $\lambda_1 \geq n - 4$. Leaving us exactly 12 cases to check.

In each of these cases, we first construct a vector v that is S_{n-4} -invariant to which we can apply σ . Then we show by explicit computation that $\sigma(v) \neq -v$. To simplify this computation, it turns out we don’t need to compute the entire invariant vector, just its “leading term,” in the following sense.

Here we follow the presentation by Peel [Pee75, Theorem 5.1] (covered nicely in Sagan’s textbook [Sag01, Theorem 2.8.3]). Namely, when we restrict an S_n -irrep V indexed by partition λ down to S_{n-1} , we define the following chain of subspaces: suppose that among standard Young tableaux, n can appear in k distinct places (these are sometimes called the inner corners of the shape λ). Let V_i be the subspace of V spanned by the Young tableaux in which n occurs among the top-most i places among those places it can occur. Then each V_i is an S_{n-1} -submodule, and V_i/V_{i-1} is the S_{n-1} -irrep indexed by the shape which is λ minus the position n was in in V_i . Now, since V_i is an S_{n-1} -submodule, note that when we apply a permutation $\pi \in S_{n-1}$ to some standard Young tableaux in V_i , we get back a linear combination of tableaux in $V_i \setminus V_{i-1}$ —which we refer to as the “leading terms”—and those in V_{i-1} —which we refer to as “lower-order terms.”

We may then iterate this to get down from S_n to S_{n-4} . From the preceding paragraph we see that in order to reach an S_{n-4} -invariant, any boxes below the first row must be filled with a subset of $\{n - 3, n - 2, n - 1, n\}$, with the remainder of that set occurring on the right-hand side of the first

row. Since our C_2 action is $(n-3, n)(n-2, n-1)$, we only focus on the position of these four. Note that if $\sigma(v) = -v$, then in particular the leading term(s) of $\sigma(v)$ must be the negative of the leading term(s) of v . So rather than compute v explicitly and the action of C_2 on it, we focus on its leading terms. We write each calculation as the presentation of the selected v followed by the C_2 action σ applied to v .

To determine the action of S_n on Young tableaux of a given shape, we use the standard “straightening rule” for the action of S_n on Young tableaux; we refer the reader to [Ful97, Section 7.4] for details.

For this calculation, we assume $n \geq 8$; we discuss this further, as well as the cases $n = 6, 7$ after the calculations:

1. $\lambda = (n)$. This is the trivial representation of S_n , so restriction to any subgroup is still trivial, and thus contains a non-zero invariant.
2. $\lambda = (n-1, 1)$.

$$v = \begin{array}{|c|} \hline n \\ \hline \end{array} \begin{array}{|c|c|c|} \hline n-3 & n-2 & n-1 \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|} \hline n-3 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline n & n-1 & n-2 \\ \hline \end{array} = \begin{array}{|c|} \hline n-3 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline n-2 & n-1 & n \\ \hline \end{array} \neq -v$$

3. $\lambda = (n-2, 2)$.

$$v = \begin{array}{|c|c|} \hline n-1 & n \\ \hline \end{array} \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|} \hline n-2 & n-3 \\ \hline \end{array} \begin{array}{|c|c|} \hline n & n-1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline \end{array} \begin{array}{|c|c|} \hline n-1 & n \\ \hline \end{array} \neq -v$$

4. $\lambda = (n-2, 1, 1)$.

$$v = \begin{array}{|c|} \hline n-1 \\ \hline n \\ \hline \end{array} \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|} \hline n-2 \\ \hline n-3 \\ \hline \end{array} \begin{array}{|c|c|} \hline n & n-1 \\ \hline \end{array} = - \begin{array}{|c|} \hline n-3 \\ \hline n-2 \\ \hline \end{array} \begin{array}{|c|c|} \hline n-1 & n \\ \hline \end{array} \neq -v$$

5. $\lambda = (n-3, 3)$.

$$v = \begin{array}{|c|c|c|} \hline n-2 & n-1 & n \\ \hline \end{array} \begin{array}{|c|} \hline n-3 \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|c|} \hline n-1 & n-2 & n-3 \\ \hline \end{array} \begin{array}{|c|} \hline n \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline n-3 & n-2 & n-1 \\ \hline \end{array} \begin{array}{|c|} \hline n \\ \hline \end{array} \neq -v$$

6. $\lambda = (n-3, 2, 1)$.

$$v = \begin{array}{|c|c|} \hline n-2 & n-1 \\ \hline n \\ \hline \end{array} \begin{array}{|c|} \hline n-3 \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|} \hline n-1 & n-2 \\ \hline n-3 \\ \hline \end{array} \begin{array}{|c|} \hline n \\ \hline \end{array} = - \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline n-1 \\ \hline \end{array} \begin{array}{|c|} \hline n \\ \hline \end{array} \neq -v$$

7. $\lambda = (n - 3, 1, 1, 1)$.

$$v = \begin{array}{|c|} \hline n-3 \\ \hline n-2 \\ \hline n-1 \\ \hline n \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|} \hline n \\ \hline n-1 \\ \hline n-2 \\ \hline n-3 \\ \hline \end{array} = - \begin{array}{|c|} \hline n \\ \hline n-3 \\ \hline n-2 \\ \hline n-1 \\ \hline \end{array} \neq -v$$

8. $\lambda = (n - 4, 4)$.

$$v = \begin{array}{|c|c|c|c|} \hline n-3 & n-2 & n-1 & n \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|c|c|} \hline n & n-1 & n-2 & n-3 \\ \hline \end{array} = - \begin{array}{|c|c|c|c|} \hline n-3 & n-2 & n-1 & n \\ \hline \end{array} \neq -v$$

9. $\lambda = (n-4, 3, 1)$. For this partition, we do not start with the standard filling, since the standard filling is sent to minus itself (plus lower-order terms, but we want to avoid having to calculate lower-order terms).

$$v = \begin{array}{|c|c|c|} \hline n-3 & n-2 & n \\ \hline n-1 & & \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|c|} \hline n & n-1 & n-3 \\ \hline n-2 & & \\ \hline \end{array} = - \begin{array}{|c|c|c|} \hline n-3 & n-2 & n-1 \\ \hline n & & \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline n-3 & n-1 & n \\ \hline n-2 & & \\ \hline \end{array} \neq -v$$

10. $\lambda = (n - 4, 2, 2)$.

$$v = \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline n-1 & n \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|} \hline n & n-1 \\ \hline n-2 & n \\ \hline \end{array} = \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline n-1 & n \\ \hline \end{array} \neq -v$$

11. $\lambda = (n - 4, 2, 1, 1)$.

$$v = \begin{array}{|c|c|} \hline n-3 & n-2 \\ \hline n-1 & \\ \hline n & \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|c|} \hline n & n-1 \\ \hline n-2 & \\ \hline n-3 & \\ \hline \end{array} = - \begin{array}{|c|c|} \hline n-3 & n-1 \\ \hline n-2 & \\ \hline n & \\ \hline \end{array} \neq -v$$

12. $\lambda = (n - 4, 1, 1, 1, 1)$.

$$v = \begin{array}{|c|} \hline n-3 \\ \hline n-2 \\ \hline n-1 \\ \hline n \\ \hline \end{array}; \quad \sigma(v) = \begin{array}{|c|} \hline n \\ \hline n-1 \\ \hline n-2 \\ \hline n-3 \\ \hline \end{array} = \begin{array}{|c|} \hline n-3 \\ \hline n-2 \\ \hline n-1 \\ \hline n \\ \hline \end{array} \neq -v$$

Note that, in order for the above calculation to not run into any “unexpected adjacencies” between the boxes, even diagonal adjacencies, it suffices for $n \geq 8$ (the worst case is the partition $(n - 3, 3)$). For $n = 6, 7$, we use Sage code to explicitly check that there are multiplicity obstructions but not occurrence obstructions; we include the code in Appendix A. This completes the proof. \square

7 Separating invariants and the Reconstruction Conjectures

In this section we give explicit invariant polynomials that identify a graph, and we show that improving the support-degree (resp. degree) of the invariants we construct is equivalent to the (Edge) Reconstruction Conjecture.

By a general theorem in invariant theory (e. g., [DK15, Thm. 2.3.6]), since the orbits of the S_n action on graphs are finite sets (or more generally Zariski-closed, but in our case we have the stronger property of actually being finite), for any two non-isomorphic graphs G, H there exists an invariant polynomial whose value on G (and its orbit) and is different than its value on H (and its orbit). In fact, there exists an invariant polynomial whose value on the orbit of G differs from its value on *all* graphs non-isomorphic to G ; we say such a polynomial *identifies* (the isomorphism class of) G . However, as we are interested in complexity-theoretic aspects of these invariant polynomials, we would like to see them explicitly. We furnish two such explicit polynomials in the next two results. We doubt these are genuinely new, but being unable to find them in the literature we include their proofs here.

We can also get a similar polynomial of slightly higher degree by using “generalized monomials” that correspond to induced (rather than general) subgraphs. The degree is higher by a factor of at least 2, but the proof is simpler, so we start there:

Proposition 7.1. *For any n -vertex graph G , there exists an invariant polynomial that identifies G , of degree N (where N is the number of edges in the complete graph on $V(G)$) and support-degree n .*

(We phrase it as above so it is clear the same proof works in the undirected case, the directed case, the bipartite case, etc., *mutatis mutandis*.)

Proof. Define

$$I_G(X) := \frac{1}{|\text{Aut}(G)|} \sum_{\sigma \in S_n} \sigma \left[\left(\prod_{(i,j) \in E(G)} x_{ij} \right) \left(\prod_{(i,j) \notin E(G)} (1 - x_{ij}) \right) \right].$$

(“ I ” is for “induced subgraph.”) Each term in the above sum is a product of variables and one-minus-variables, with each variable occurring in exactly one factor of the product; it is readily apparent that such a term is 1 on exactly one graph and 0 on all other graphs (not just up to isomorphism). And since I_G is a sum over all isomorphic copies, we get $I_G(H) = 1$ if and only if $H \cong G$, and otherwise $I_G(H) = 0$. \square

Using the S_G polynomials as a building block instead of I_G , we can improve the degree by a factor of 2, with a slightly more complicated proof.

Proposition 7.2. *For any graph G , there exists:*

1. *an invariant polynomial that identifies G and has degree at most $\max\{2, \min\{|E(G)|, |E(G^c)|\}\}$*
2. *an invariant polynomial that identifies G and, using $V_0(G)$ to denote the set of non-isolated vertices of G , has support-degree at most $\max\{\min\{|V(G)|, 4\}, \min\{|V_0(G)|, |V_0(G^c)|\}\}$.*

Proof. 1. Suppose without loss of generality that the number of edges of G is less than the number of non-edges (if it’s not, apply the ring automorphism that effectively takes the complement of graphs: $x_{ij} \leftrightarrow 1 - x_{ij}$ for all (i, j)). Then we claim the following polynomial satisfies the conclusion of the proposition:

$$f(X) := -1 + S_G(X) + 2 \left(|E(G)| - \sum_{i,j \in [n]} x_{ij} \right)^2.$$

(Note that here $|E(G)|$ is an integer constant, independent of the input variables to f .) We have already seen that S_G is invariant; the other summand is the square of a polynomial that is readily seen to be invariant, being the sum of a constant and *all* the variables, with the same coefficient.

Now, if $H \cong G$, then we have

$$\begin{aligned} f(H) &= -1 + S_G(H) + 2(|E(G)| - \sum x_{ij}(H))^2 \\ &= -1 + 1 + 2(|E(G)| - |E(H)|)^2 = 0. \end{aligned}$$

On the other hand, suppose H is any n -vertex graph that is not isomorphic to G . If H contains an isomorphic copy of G as a subgraph, then since $H \not\cong G$, G must be a proper subgraph of H , so $|E(H)| > |E(G)|$. Thus the sum inside the second term is a nonzero integer, so its square is at least 1. We then have:

$$\begin{aligned} f(H) &= -1 + S_G(H) + 2(|E(G)| - \sum x_{ij}(H))^2 \\ &= -1 + 1 + 2(|E(G)| - |E(H)|)^2 \\ &\geq 0 + 2 \cdot 1 \geq 2, \end{aligned}$$

so in particular $f(H) \neq 0$.

On the other hand, if $|E(H)| = |E(G)|$, then since $H \not\cong G$, H cannot contain G as a subgraph, and in that case we have:

$$\begin{aligned} f(H) &= -1 + S_G(H) + 2(|E(G)| - \sum x_{ij}(H))^2 \\ &= -1 + 0 + 2(|E(G)| - |E(H)|)^2 \\ &= -1 + 0 + 0 = -1 \neq 0. \end{aligned}$$

Thus, for any $H \not\cong G$, we have $f(H) \neq 0$.

2. The same construction yields the statement on support-degree, but now we choose G or its complement based on which one has fewer non-isolated vertices. The 4 comes from the fact that $(|E(G)| - \sum x_{ij})^2$ includes terms of the form $x_{ij}x_{k\ell}$. □

Whether or not the degree bounds in the preceding results can be improved is a question closely related to (standard generalizations of) various Reconstruction Conjectures. We next recall these conjectures and then make this connection explicit.

The k -vertex deck of a graph G is the multiset of isomorphism types gotten by removing k vertices from G : $VD_k(G) := \{[G \setminus U] : U \in \binom{V(G)}{k}\}$, where we use $[G]$ for the isomorphism type; similarly, the k -edge deck of a graph is $ED_k(G) := \{[G \setminus E] : E \in \binom{E(G)}{k}\}$. Given some isomorphism-invariant function f on the set of graphs, a graph G is said to be *reconstructible from f* if $f(G) = f(H)$ implies $G \cong H$. In this language, we have various reconstruction conjectures:

Conjecture 7.3 (Reconstruction Conjecture [Kel42, Kel57, Ula60]). Every finite simple undirected graph on at least 3 vertices is reconstructible from its 1-vertex deck.

Conjecture 7.4 (Edge-Reconstruction Conjecture [Har64]). Every finite simple undirected graph with at least 4 edges is reconstructible from its 1-edge deck.

As recently as 2020, Farhadian proved that a random graph is reconstructible from the multiset of its $3 \log n$ -vertex subgraphs with high probability [Far20] (in the above language, reconstructible from its $(n - 3 \log n)$ -deck). On the other hand, for any $0 < \varepsilon < 1$, there exist arbitrarily large pairs of non-isomorphic graphs that have identical multisets of $\lfloor \varepsilon n \rfloor$ -vertex subgraphs [Nýd92]. A directed version of the Reconstruction Conjecture turned out to be false [Sto77], but there is a so-called New Digraph

Reconstruction Conjecture that includes for each vertex v the triple $([G \setminus v], \deg_{in}(v), \deg_{out}(v))$ [Ram81]. We are not aware of work generalizing the New Digraph Reconstruction Conjecture to removing more than 1 vertex. For complexity-theoretic aspects of these conjectures and further references, see [HHRT07].

Theorem 7.5 (Relating invariants and edge reconstruction, cf. [For04, Thm. 4.6]). *Let k be a positive integer, and let G be a finite simple undirected graph with $k < |E(G)|$. Then G is k -edge reconstructible if and only if it is identified by invariant polynomials of degree at most $|E(G)| - k$.*

This theorem is essentially equivalent to Forman’s Theorem 4.6, though we discovered the result independently before becoming aware of Forman’s paper; see Section 1.1 for a comparison.

Proof. Suppose G is identified by invariant polynomials of degree at most $|E(G)| - k$. Then for each graph H on the same number of vertices as G , there is some graph K with $|E(K)| \leq |E(G)| - k$ such that $S_K(G) \neq S_K(H)$ (for otherwise all invariants of degree at most $|E(G)| - k$ take the same values on both G and H , by Observation 3.10). Since the value of $S_K(G)$ is the same as the multiplicity of K as an edge-deleted subgraph of G , and the counts of k -edge-deleted subgraphs are determined by $ED_k(G)$, we have that for each graph H not isomorphic to G , $ED_k(G) \neq ED_k(H)$, and thus G is k -edge-reconstructible.

Conversely, suppose G is k -edge-reconstructible. Then for every graph H on the same number of vertices that is not isomorphic to G , we have $ED_k(G) \neq ED_k(H)$. In particular, there exists a graph K with $|E(K)| \leq |E(G)| - k$ whose multiplicity in $ED_k(G)$ differs from its multiplicity in $ED_k(H)$. These multiplicities are precisely $S_K(G)$ and $S_K(H)$, respectively, so we get that $S_K(G) \neq S_K(H)$. Thus the invariant polynomial $S_K(X) - S_K(G)$ vanishes on G but not on H , so G is identified by invariants of degree at most $|E(G)| - k$. \square

Theorem 7.6 (Relating invariants and vertex reconstruction, cf. [For04, Theorem 11.4]). *Let k be a positive integer, and let G be a finite simple undirected graph with $k < |V(G)|$. Then G is k -vertex reconstructible if and only if it is identified by invariant polynomials of support-degree at most $|V(G)| - k$.*

Proof. Suppose G is identified by invariant polynomials of support-degree at most $|V(G)| - k$. Then for each graph H on the same number of vertices as G , there is some graph K with $|V(K)| \leq |V(G)| - k$ such that $S_K(G) \neq S_K(H)$, for otherwise all invariants of support-degree at most $|V(G)| - k$ take the same values on both G and H , by Observation 3.10. By Möbius inversion, the census of non-induced subgraphs on $\leq |V(G)| - k$ vertices and the census of induced subgraphs on $\leq |V(G)| - k$ vertices determine one another. But the latter is precisely $VD_k(G)$. Thus we have that for each graph H not isomorphic to G , $VD_k(G) \neq VD_k(H)$, and thus G is k -reconstructible.

Conversely, suppose G is k -reconstructible. Then for every graph H on the same number of vertices that is not isomorphic to G , we have $VD_k(G) \neq VD_k(H)$. In particular, there exists a graph K with $|V(K)| \leq |V(G)| - k$ whose multiplicity in $VD_k(G)$ differs from its multiplicity in $VD_k(H)$. By what we’ve said above, there must be some graph K' on $\leq |V(G)| - k$ vertices such that $S_{K'}(G) \neq S_{K'}(H)$, and thus G is identified by invariants of support-degree at most $|V(G)| - k$. \square

Acknowledgment

We thank Benedikt Pago for pointing us to [DPS26]—in an earlier version of this paper we had posed an “open” question that it turned out they had already answered! JU thanks Anuj Dawar for the invitation to speak at Cambridge, at which the aforementioned conversation occurred. We thank Sandra Kiefer for conversations related to graphs characterized by their symmetries. JAG thanks Emma Church for teaching him about representation stability and FI-modules, Jonah Blasiak for useful conversations over several years, and Maria Gillespie for interesting related conversations

about representations of the symmetric group. Initial parts of this project were carried out as JU’s master’s research at U. Massachusetts, Amherst; JU thanks Cameron Musco for being his master’s advisor. JAG was partially funded through NSF CAREER award CCF-2047756; JU was partially funded by Grochow sabbatical funds and a fellowship from the University of Colorado Boulder Department of Computer Science.

AI Usage Statement

To the best of our knowledge, no LLM was used in the research, writing, nor any other aspect of this paper.

A Code for the calculation in Proposition 6.11

We used the following code in SageMath version 10.7 on 22 June 2026. This code constructs the two graphs G_n, H_n from the proof of Proposition 6.11, then retrieves their multiplicity information as vectors—using the connection with cycle indices as in the proof of Theorem 6.5—and checks that there are no occurrence obstructions.

```
def graph_to_multiplicity_vec(G,n):
    Sn = SymmetricGroup(n)
    group = G.automorphism_group()

    # gets a list of cycle types in the order of columns of the character table
    cycle_types = [g.cycle_type()
                    for g in Sn.conjugacy_classes_representatives()]

    polynomial = list(group.cycle_index())
    # NB: .cycle_index() returns a polynomial which is already
    # divided by the order of the group (works with our proof of the theorem)

    vec = [0]*len(cycle_types) # list of 0s of length len(cycle_types)
    for monomial in polynomial:
        vec[cycle_types.index(monomial[0])] = monomial[1] # mon[0]==term, mon[1]==coeff
    M = Sn.character_table()
    return(M*vector(vec)) # multiply character table by normalized cycle index vector

n = 6 # number of vertices; replace as needed

# dandelion graph
H = Graph({0:[1],1:[2],2:[3],3:[i for i in range(4,n)]})

# dandelion with seeds blown off
G = Graph({0:[1],1:[2],2:[3]})
for i in range(4,n):
    G.add_vertex()

G_mult = graph_to_multiplicity_vec(G,n)
H_mult = graph_to_multiplicity_vec(H,n)

G.show()
```

```

print(G_mult)
H.show()
print(H_mult)

for i in range(len(G_mult)):
    if (G_mult[i] == 0) != (H_mult[i] == 0):
        print(False) # this line should never be reached

```

References

- [AD17a] Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Inf. Comput.*, 256:2–8, 2017. doi:10.1016/j.ic.2017.04.004.
- [AD17b] Matthew Anderson and Anuj Dawar. On symmetric circuits and fixed-point logics. *Theory Comput. Syst.*, 60(3):521–551, 2017. Originally appeared in **STACS 2014**; preprint arXiv:1401.1125. doi:10.1007/S00224-016-9692-2.
- [AM13] Albert Atserias and Elitza N. Maneva. Sherali–Adams relaxations and indistinguishability in counting logics. *SIAM J. Comput.*, 42(1):112–137, 2013. doi:10.1137/120867834.
- [AYZ95] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. Assoc. Comput. Mach.*, 42(4):844–856, 1995. Preliminary version appeared in **STOC '94**. doi:10.1145/210332.210337.
- [Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016. Preprint arXiv:1512.03547. doi:10.1145/2897518.2897542.
- [BG15] Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2015. Preprint arXiv:1502.05912. doi:10.1007/978-3-662-47672-7_13.
- [BG17] Christoph Berkholz and Martin Grohe. Linear Diophantine equations, group CSPs, and graph isomorphism. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 327–339. SIAM, 2017. Preprint arXiv:1607.04287 [cs.CC]. doi:10.1137/1.9781611974782.21.
- [BGS99] Andreas Blass, Yuri Gurevich, and Saharon Shelah. Choiceless polynomial time. *Ann. Pure Appl. Log.*, 100(1-3):141–187, 1999. doi:10.1016/S0168-0072(99)00005-6.
- [BIP19] Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. *J. Am. Math. Soc.*, 32(1):163–193, 2019. Originally appeared in **FOCS 2016**; Preprint arXiv:1604.06431. doi:10.1090/jams/908.
- [BL08] Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *J. Algebra*, 320(11):4020–4029, 2008. doi:10.1016/j.jalgebra.2008.07.014.

- [CEF15] Thomas Church, Jordan S. Ellenberg, and Benson Farb. FI-modules and stability for representations of symmetric groups. *Duke Math. J.*, 164(9):1833–1910, 2015. doi:10.1215/00127094-3120274.
- [CF13] Thomas Church and Benson Farb. Representation theory and homological stability. *Adv. Math.*, 245:250–314, 2013. doi:10.1016/j.aim.2013.06.016.
- [CFI92] Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Comb.*, 12(4):389–410, 1992. Originally appeared in FOCS 1989. doi:10.1007/BF01305232.
- [CIK97] Alexander L. Chistov, Gábor Ivanyos, and Marek Karpinski. Polynomial time algorithms for modules over finite dimensional algebras. In Bruce W. Char, Paul S. Wang, and Wolfgang Küchlin, editors, *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, ISSAC 1997, Maui, Hawaii, USA, July 21-23, 1997*, pages 68–74. ACM, 1997. doi:10.1145/258726.258751.
- [CKW10] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Found. Trends Theor. Comput. Sci.*, 6(1-2):1–138, 2010. doi:10.1561/04000000043.
- [Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.
- [Der13] Harm Derksen. The Graph Isomorphism Problem and approximate categories. *J. Symb. Comput.*, 59:81–112, 2013. doi:10.1016/J.JSC.2013.06.002.
- [DF04] David S. Dummit and Richard M. Foote. *Abstract algebra*. Chichester: Wiley, 3rd ed. edition, 2004.
- [DFL⁺21] Neta Dafni, Yuval Filmus, Noam Lifshitz, Nathan Lindzey, and Marc Vinyals. Complexity measures on the symmetric group and beyond (extended abstract). In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, Virtual Conference, January 6-8, 2021*, volume 185 of *LIPICs*, pages 87:1–87:5. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. Preprint arXiv:2010.07405. doi:10.4230/LIPICs.ITCS.2021.87.
- [DGKP25] Anuj Dawar, Erich Grädel, Leon Kullmann, and Benedikt Pago. Symmetric Proofs in the Ideal Proof System. In Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:18, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.MFCS.2025.40.
- [DIP20] Julian Dörfler, Christian Ikenmeyer, and Greta Panova. On geometric complexity theory: multiplicity obstructions are stronger than occurrence obstructions. *SIAM J. Appl. Algebra Geom.*, 4(2):354–376, 2020. Originally appeared in ICALP 2019; Preprint arXiv:1901.04576. doi:10.1137/19M1287638.
- [DK15] Harm Derksen and Gregor Kemper. *Computational invariant theory. With two appendices by Vladimir L. Popov and an addendum by Nobert A. Campo and Vladimir L. Popov*, volume 130 of *Encycl. Math. Sci.* Berlin: Springer, 2nd enlarged edition edition, 2015. doi:10.1007/978-3-662-48422-7.

- [DPS26] Anuj Dawar, Benedikt Pago, and Tim Seppelt. Symmetric algebraic circuits and homomorphism polynomials. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference, ITCS 2026, Bocconi University, Milan, Italy, January 27-30, 2026*, volume 362 of *LIPICs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026. Preprint of full version at [arXiv:2502.06740](https://arxiv.org/abs/2502.06740). doi:10.4230/LIPICs.ITCS.2026.46.
- [DW22] Anuj Dawar and Gregory Wilsenach. Symmetric circuits for rank logic. *ACM Trans. Comput. Log.*, 23(1):6:1–6:35, 2022. Originally appeared in [CSL 2018](#). doi:10.1145/3476227.
- [DW25] Anuj Dawar and Gregory Wilsenach. Symmetric arithmetic circuits. *Theory of Computing*, 21(14):1–32, 2025. Originally appeared in [ICALP 2020](#); preprint [arXiv:2002.06451](https://arxiv.org/abs/2002.06451). doi:10.4086/toc.2025.v021a014.
- [EFP11] David Ellis, Ehud Friedgut, and Haran Pilpel. Intersecting families of permutations. *J. Am. Math. Soc.*, 24(3):649–682, 2011. Corrected version on arXiv [1011.3342](https://arxiv.org/abs/1011.3342). doi:10.1090/S0894-0347-2011-00690-5.
- [ER63] Pál Erdős and Alfréd Rényi. Asymmetric graphs. *Acta Math. Acad. Sci. Hung.*, 14:295–315, 1963. doi:10.1007/BF01895716.
- [Far20] Ameneh Farhadian. Almost every n -vertex graph is determined by its $3\log_2 n$ -vertex subgraphs. *Int. J. Found. Comput. Sci.*, 31(5):611–619, 2020. doi:10.1142/S012905412050029X.
- [FH91] William Fulton and Joe Harris. *Representation theory. A first course*, volume 129 of *Grad. Texts Math.* New York etc.: Springer-Verlag, 1991. doi:10.1007/978-1-4612-0979-9.
- [For04] Robin Forman. Finite-type invariants for graphs and graph reconstructions. *Adv. Math.*, 186(1):181–228, 2004. doi:10.1016/j.aim.2003.08.013.
- [Ful97] William Fulton. *Young tableaux. With applications to representation theory and geometry*, volume 35 of *Lond. Math. Soc. Stud. Texts*. Cambridge: Cambridge University Press, 1997. doi:10.1017/CB09780511626241.
- [Für01] Martin Fürer. Weisfeiler–Lehman refinement requires at least a linear number of iterations. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, volume 2076 of *Lecture Notes in Computer Science*, pages 322–333. Springer, 2001. doi:10.1007/3-540-48224-5_27.
- [GC05] Alain Goupil and Cedric Chauve. Combinatorial operators for Kronecker powers of representations of \mathfrak{S}_n . *Sémin. Lothar. Comb.*, 54:b54j, 13, 2005. URL: <https://eudml.org/doc/228778>.
- [GQ23] Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: Tensor Isomorphism-completeness. *SIAM J. Comput.*, 52:568–617, 2023. Part of the preprint [arXiv:1907.00309 \[cs.CC\]](https://arxiv.org/abs/1907.00309). Preliminary version appeared at [ITCS '21](#). doi:10.1137/21M1441110.
- [Gro12a] Joshua A. Grochow. Matrix Lie algebra isomorphism. In *IEEE Conference on Computational Complexity (CCC12)*, pages 203–213, 2012. Preprint of full version [arXiv:1112.2012](https://arxiv.org/abs/1112.2012), ECCC Technical Report [TR11-168](#), Ch. 4 of [\[Gro12b\]](#). doi:10.1109/CCC.2012.34.

- [Gro12b] Joshua A. Grochow. *Symmetry and equivalence relations in classical and geometric complexity theory*. PhD thesis, University of Chicago, Chicago, IL, 2012. <https://home.cs.colorado.edu/~jgrochow/grochow-thesis.pdf>.
- [Gro15] Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Comput. Complexity*, 24(2):393–475, 2015. Originally appeared in CCC '14. doi:10.1007/s00037-015-0103-x.
- [GV06] Martin Grohe and Oleg Verbitsky. Testing graph isomorphism in parallel by playing a game. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, volume 4051 of *Lecture Notes in Computer Science*, pages 3–14. Springer, 2006. Preprint [arXiv:cs/0603054](https://arxiv.org/abs/cs/0603054). doi:10.1007/11786986_2.
- [Har64] Frank Harary. On the reconstruction of a graph from a collection of subgraphs. In *Theory Graphs Appl., Proc. Symp. Smolenice 1963, 47-52 (1964)*. Publ. House Czechoslovak Acad. Sci., Prague, 1964.
- [HHRT07] Edith Hemaspaandra, Lane A. Hemaspaandra, Stanisław P. Radziszowski, and Rahul Tripathi. Complexity results in graph reconstruction. *Discrete Appl. Math.*, 155(2):103–118, 2007. Originally appeared in MFCS '04. doi:10.1016/j.dam.2006.04.038.
- [JK09] Gordon James and Adalbert Kerber. *The representation theory of the symmetric group*. Foreword by P. M. Cohn, introduction by G. de B. Robinson., volume 16 of *Encycl. Math. Appl.* Cambridge: Cambridge University Press, reprint of the 1985 hardback ed. edition, 2009. doi:10.1017/CB09781107340732.
- [Kay15] Neeraj Kayal. Topics in complexity theory, lecture 5. https://www.csa.iisc.ac.in/~chandan/courses/arithmetric_circuits/notes/lec5.pdf, 2015. Scribed by Sumant Heyde; Spring 2015 course run with Chandan Saha.
- [Kel42] Paul J. Kelly. *On isometric transformations*. PhD thesis, University of Wisconsin–Madison, 1942.
- [Kel57] Paul J. Kelly. A congruence theorem for trees. *Pac. J. Math.*, 7:961–968, 1957. doi:10.2140/pjm.1957.7.961.
- [Koc82] W. L. Kocay. Some new methods in reconstruction theory. In *Combinatorial mathematics IX, Proc. 9th Aust. Conf., Brisbane/ Aust. 1981, Lect. Notes Math. 952, 89-114 (1982)*. 1982. doi:10.1007/bfb0061974.
- [KSV02] Jeong Han Kim, Benny Sudakov, and Van H. Vu. On the asymmetry of random regular graphs and random graphs. *Random Struct. Algorithms*, 21(3-4):216–224, 2002. doi:10.1002/rsa.10054.
- [Lev73] Leonid A. Levin. Universal search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.
- [Lie87] Martin W. Liebeck. The affine permutation groups of rank three. *Proc. Lond. Math. Soc. (3)*, 54:477–516, 1987. doi:10.1112/plms/s3-54.3.477.
- [LW25] Vladimir Lysikov and Michael Walter. Complexity theory of orbit closure intersection for tensors: reductions, completeness, and graph isomorphism hardness. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 486–511. IEEE, 2025. Preprint [arXiv:2411.04639](https://arxiv.org/abs/2411.04639). doi:10.1109/FOCS63196.2025.00027.

- [Mar10] Dániel Marx. Can you beat treewidth? *Theory Comput.*, 6:85–112, 2010. Id/No 5. doi:10.4086/toc.2010.v006a005.
- [MS01] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. I: An approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001. doi:10.1137/S009753970038715X.
- [Neu24] Daniel Neuen. Homomorphism-distinguishing closedness for graphs of bounded tree-width. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, Clermont-Ferrand, France, March 12-14, 2024*, LIPIcs, pages 53:1–53:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. Preprint of full version arXiv:2304.07011. doi:10.4230/LIPICS.STACS.2024.53.
- [Nýd92] Václav Nýdl. Finite undirected graphs which are not reconstructible from their large cardinality subgraphs. *Discrete Math.*, 108(1-3):373–377, 1992. doi:10.1016/0012-365X(92)90689-D.
- [Pee75] M. H. Peel. Specht modules and symmetric groups. *J. Algebra*, 36:88–97, 1975. doi:10.1016/0021-8693(75)90158-1.
- [Pou77] Maurice Pouzet. Quelques remarques sur les résultats de Tutte concernant le problème de Ulam. *Publ. Dép. Math., Lyon*, 14(2):1–8, 1977.
- [Pou79] Maurice Pouzet. Note sur le problème de Ulam. *J. Comb. Theory, Ser. B*, 27:231–236, 1979. doi:10.1016/0095-8956(79)90015-7.
- [PT01] Maurice Pouzet and Nicolas M. Thiéry. Algebraic graph invariants and reconstruction. *C. R. Acad. Sci., Paris, Sér. I, Math.*, 333(9):821–826, 2001. doi:10.1016/S0764-4442(01)02137-1.
- [Ram81] S. Ramachandran. On a new digraph reconstruction conjecture. *J. Comb. Theory, Ser. B*, 31:143–149, 1981. doi:10.1016/S0095-8956(81)80019-6.
- [RRR19] C. Ramya and B. V. Raghavendra Rao. Linear projections of the Vandermonde polynomial. *Theor. Comput. Sci.*, 795:165–182, 2019. doi:10.1016/j.tcs.2019.06.010.
- [Sag01] Bruce E. Sagan. *The symmetric group. Representations, combinatorial algorithms, and symmetric functions.*, volume 203 of *Grad. Texts Math.* New York, NY: Springer, 2nd ed. edition, 2001. doi:10.1007/978-1-4757-6804-6.
- [SSC14] Aaron Snook, Grant Schoenebeck, and Paolo Codenotti. Graph Isomorphism and the Lasserre hierarchy. arXiv:1401.0758 [cs.CC], 2014.
- [Sto77] Paul K. Stockmeyer. The falsity of the reconstruction conjecture for tournaments. *J. Graph Theory*, 1:19–25, 1977. URL: scholarworks.wm.edu/aspubs/1120, doi:10.1002/jgt.3190010108.
- [Thi00] Nicolas M. Thiéry. Algebraic invariants of graphs; a study based on computer exploration. *SIGSAM Bull.*, 34(3):9–20, 2000. Preprint arXiv:0812.3082. doi:10.1145/377604.377612.
- [TW23] Jacobo Torán and Florian Wörz. Number of variables for graph differentiation and the resolution of graph isomorphism formulas. *ACM Trans. Comput. Log.*, 24(3):23:1–23:25, 2023. Originally appeared in [CSL 2022](#); ECCC preprint [TR21-097](#). doi:10.1145/3580478.

- [Ula60] S. M. Ulam. *A collection of mathematical problems*, volume 8 of *Intersci. Tracts Pure Appl. Math.* Interscience Publishers, New York, NY, 1960.
- [WL68] B. Weisfeiler and A. Leman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, Ser. 2(N9):12–16, 1968.