

Computational Complexity of the Weisfeiler-Leman Dimension

Moritz Lichter ✉ 

RWTH Aachen University, Germany

Simon Raßmann ✉ 

TU Darmstadt, Germany

Pascal Schweitzer ✉ 

TU Darmstadt, Germany

Abstract

The Weisfeiler-Leman dimension of a graph G is the least number k such that the k -dimensional Weisfeiler-Leman algorithm distinguishes G from every other non-isomorphic graph, or equivalently, the least k such that G is definable in $(k + 1)$ -variable first-order logic with counting. The dimension is a standard measure of the descriptive or structural complexity of a graph and recently finds various applications in particular in the context of machine learning. This paper studies the complexity of computing the Weisfeiler-Leman dimension. We observe that deciding whether the Weisfeiler-Leman dimension of G is at most k is NP-hard, even if G is restricted to have 4-bounded color classes. For each fixed $k \geq 2$, we give a polynomial-time algorithm that decides whether the Weisfeiler-Leman dimension of a given graph with 5-bounded color classes is at most k . Moreover, we show that for these bounds on the color classes, this is optimal because the problem is P-hard under logspace-uniform AC_0 -reductions. Furthermore, for each larger bound c on the color classes and each fixed $k \geq 2$, we provide a polynomial-time decision algorithm for the abelian case, that is, for structures of which each color class has an abelian automorphism group.

While the graph classes we consider may seem quite restrictive, graphs with 4-bounded abelian colors include CFI-graphs and multipedes, which form the basis of almost all known hard instances and lower bounds related to the Weisfeiler-Leman algorithm.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity theory and logic

Keywords and phrases Weisfeiler-Leman algorithm, dimension, complexity, coherent configurations

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.13

Related Version *Full Version:* <https://arxiv.org/abs/2402.11531> [35]

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (EngageS: grant agreement No. 820148).

Moritz Lichter: The research of this author has received further funding by the European Union (ERC, SymSim, 101054974).

1 Introduction

The *Weisfeiler-Leman algorithm* is a simple combinatorial procedure studied in the context of the graph isomorphism problem. For every $k \geq 1$, the algorithm has a k -dimensional variant, k -WL for short, that colors k -tuples of vertices according to how they structurally sit inside the whole graph: if two tuples get different colors, they cannot be mapped onto each other by an automorphism of the graph (while the converse is not always true). The 1-dimensional algorithm, which is also known as *color refinement*, starts by coloring each



© Moritz Lichter, Simon Raßmann, and Pascal Schweitzer;

licensed under Creative Commons License CC-BY 4.0

33rd EACSL Annual Conference on Computer Science Logic (CSL 2025).

Editors: Jörg Endrullis and Sylvain Schmitz; Article No. 13; pp. 13:1–13:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

vertex according to its degree, and then repeatedly refines this coloring by including into each vertex color the multisets of colors of its neighbors. The k -dimensional variant generalizes this idea and colors k -tuples of vertices instead of single vertices [43, 11].

The Weisfeiler-Leman algorithm plays an important role in both theoretic and practical approaches to the graph isomorphism problem, but is also related to a plethora of seemingly unrelated areas: to finite model theory and descriptive complexity via the correspondence of k -WL to $(k + 1)$ -variable first-order logic with counting [11, 27], to machine learning via a correspondence to the expressive power of (higher-dimensional) graph neural networks [37], to the Sherali-Adams hierarchy in combinatorial optimization [3, 25], and to homomorphism counts from treewidth- k graphs [14]. On the side of practical graph isomorphism, the color refinement procedure is a basic building block of the so-called *individualization-refinement framework*, which is the basis of almost every modern practical solver for the graph isomorphism problem [36, 28, 29, 1]. On the side of theoretical graph isomorphism, Babai's quasipolynomial-time algorithm for the graph isomorphism problem [4] uses a combination of group-theoretic techniques and a logarithmic-dimensional Weisfeiler-Leman algorithm.

The Weisfeiler-Leman algorithm is a powerful algorithm for distinguishing non-isomorphic graphs on its own. For every k , k -WL can be used as an incomplete polynomial-time isomorphism test: if the multiset of colors of k -tuples of two graphs G and H differ, then G and H cannot be isomorphic. In this case, k -WL distinguishes G and H , otherwise G and H are k -WL-equivalent. For a given graph G , we say that the k -dimensional Weisfeiler-Leman algorithm k -WL *identifies* G if it distinguishes G from every non-isomorphic graph. The smallest such k is known as the *Weisfeiler-Leman dimension of G* [21].

It is known that almost all graphs have Weisfeiler-Leman dimension 1 [5]. However, color refinement fails spectacularly on regular graphs, where it always returns the monochromatic coloring. For these, it is known that 2-WL identifies almost all regular graphs [10, 32]. In contrast to these positive results, for every k there is some graph G (even of order linear in k , of maximum degree 3, and with 4-bounded abelian color classes, i.e., such that no more than 4 vertices can share the same vertex-color, and every color class induces a graph with abelian automorphism group) that is not identified by k -WL [11]. These so-called CFI-graphs have high Weisfeiler-Leman dimension and are thus hard instances for combinatorial approaches to the graph isomorphism problem.

The situation changes for restricted classes of graphs. If the Weisfeiler-Leman dimension over some class of graphs is bounded by k , then the k -WL correctly decides isomorphism over this class. And since k -WL can be implemented in polynomial time $O(n^{k+1} \log n)$ [27], this puts graph isomorphism over such classes into polynomial time. Examples of graph classes with bounded Weisfeiler-Leman dimension include graphs of bounded tree-width [23], graphs of bounded rank-width [24], graphs with 3-bounded color classes [27], planar graphs [30], and more generally every non-trivial minor-closed graph class [20].

In this paper, we study the computational complexity of computing the Weisfeiler-Leman dimension. We call the problem of deciding whether the Weisfeiler-Leman dimension of a given graph is at most k the *k -WL-identification problem*. For upper complexity bounds, non-identification of a graph G can be witnessed by providing a graph H that is not distinguished from G by k -WL but is also not isomorphic to G . As the latter can be checked in **co-NP**, this places the identification problem into the class Π_2^P of the polynomial hierarchy. If the graph isomorphism problem is solvable in polynomial time, this complexity bound collapses to **co-NP**. However, there is no apparent reason why the identification problem should not be polynomial-time decidable.

On the side of lower complexity bounds, the 1-WL-identification problem is complete for polynomial time under uniform reductions in the circuit complexity class AC_0 [31, 2]. Hardness of the 1-WL-identification problem does, however, not easily imply any hardness results for the k -WL-identification problem for higher values of k . Indeed, no hardness results are known for $k \geq 2$. The 2-WL-identification problem in particular includes the problem of deciding whether a given strongly regular graph is determined up to isomorphism by its parameters, which is a baffling problem from classic combinatorics far beyond our current knowledge. To understand the difficulties of the k -WL-identification problem better, we can again consider classes of graphs. On every class of graphs with bounded color classes, graph isomorphism is solvable in polynomial time [6, 16], which puts the identification problem over this class into co-NP for every $k \geq 2$. Graphs with 3-bounded color classes are identified by 2-WL [27], which makes their identification problem trivial. As shown by the CFI-graphs [11], this is no longer true for graphs with 4-bounded color classes. Nevertheless, as shown by Fuhlbrück, Köbler, and Verbitsky, identification of graphs with 5-bounded color classes by 2-WL is efficiently decidable [15]. For higher dimensions or bounds on the color classes essentially nothing is known.

Contribution. We extend the results of [15] from 2-WL to k -WL and give a polynomial-time algorithm deciding whether a graph with 5-bounded color classes is identified by k -WL:

► **Theorem 1.** *For every k , there is an algorithm that decides the k -WL-identification problem for vertex- and edge-colored, directed graphs with 5-bounded color classes in time $O_k(n^{O(k)})$. If such a graph G is not identified by k -WL, the algorithm provides a witness for this, i.e., a graph H that is not isomorphic to G and not distinguished from G by k -WL.*

Via the correspondence of k -WL to $(k + 1)$ -variable counting logic, Theorem 1 implies that definability of graphs with 5-bounded color classes in this logic is decidable in polynomial time. While the restriction to 5-bounded color classes may seem stark, almost all known hardness results and lower bounds for the Weisfeiler-Leman algorithm remain true for graphs with bounded color classes and in most cases even 4-bounded color classes suffice [19, 13, 40, 39, 38].

Towards generalizing Theorem 1 to arbitrary relational structures and larger color classes, we consider structures with abelian color classes, i.e., structures of which each color class induces a structure with an abelian automorphism group. Such structures were previously considered in the context of descriptive complexity theory [44], and include both CFI-graphs [11] and multipedes [39, 38] over ordered base graphs, which form the basis of all known constructions of graphs with high Weisfeiler-Leman dimension. For many cases in descriptive complexity theory, restricting to 4-bounded abelian color classes is sufficient, but in some cases larger (but still abelian) color classes are required [26, 18, 33, 34]. For such structures, we obtain a polynomial-time algorithm as before:

► **Theorem 2.** *For every $k \in \mathbb{N}$ and $c, r \leq k$, there is an algorithm that decides the k -WL-identification problem for r -ary relational structures with c -bounded abelian color classes in time $O_k(n^{O(k)})$. If such a structure \mathfrak{A} is not identified by k -WL, the algorithm provides a witness for this, i.e., a second structure \mathfrak{B} that is not isomorphic to \mathfrak{A} and not distinguished from \mathfrak{A} by k -WL.*

On the side of hardness results, we first prove that when the dimension k is part of the input, the identification problem is NP-hard. Note that a similar result was recently independently observed by Seppelt [42].

► **Theorem 3.** *The problem of deciding, given a graph G and a natural number k , whether the Weisfeiler-Leman dimension of G is at most k is NP-hard, both over uncolored simple graphs, and over simple graphs with 4-bounded color classes.*

Furthermore, we extend the P-hardness results for 1-WL [2] to arbitrary k and prove that, when k is fixed, the k -WL-identification problem is hard for polynomial time:

► **Theorem 4.** *For every $k \geq 1$, the k -WL-identification problem is P-hard under uniform AC_0 -reductions over both uncolored simple graphs, and simple graphs with 4-bounded abelian color classes.*

Techniques. To prove Theorem 1, we exploit the close connection between the coloring computed by k -WL and certain combinatorial structures called k -ary coherent configurations. These structures come with two notions of isomorphisms, algebraic ones and combinatorial ones. Similarly to [15], we reduce the k -WL-identification problem to the separability problem for k -ary coherent configurations, that is, to decide whether algebraic and combinatorial isomorphisms for a given k -ary coherent configuration coincide. We make two crucial observations: First, we show that the k -ary coherent configurations obtained from graphs are fully determined by their underlying 2-ary configurations. We call such configurations 2-induced. Second, we reduce the separability problem for arbitrary k -ary coherent configurations to the same problem on the structurally simpler class of star-free k -ary coherent configurations. Combining both observations, we show that two 2-induced, star-free k -ary coherent configurations obtained from k -WL-equivalent graphs must be isomorphic. Given such a k -ary coherent configuration obtained from a graph, it thus suffices to decide whether there is another non-isomorphic graph yielding the same configuration. Finally, we solve this problem by encoding it into the graph isomorphism problem for structures with bounded color classes, which is polynomial-time solvable [6, 16].

The main obstacle to generalize Theorem 1 to larger color classes or relational structures of higher arity is the existence of k -WL-equivalent structures that yield non-isomorphic star-free k -ary coherent configurations, which greatly increases the space of possibly equivalent but non-isomorphic structures. To make up for this, we consider structures with abelian color classes. Using both the bijective pebble game [26] and ideas from the theory of coherent configurations, we provide structural insights for the class of k -ary coherent configurations with abelian fibers. This allows us to finally prove that in the abelian case, it does suffice to consider other relational structures yielding the same k -ary coherent configuration.

NP-hardness in Theorem 3 is proved by combining the known relationship between the Weisfeiler-Leman dimension of CFI-graphs [11] and the tree-width of the underlying base graphs with the recent result that computing the tree-width of cubic graphs is NP-hard [9]. With the same techniques, we can also prove that deciding k -WL-equivalence of graphs is co-NP-hard when the dimension k is considered part of the input.

For the P-hardness result of the k -WL-identification problem in Theorem 4, we adapt a construction by Grohe [19] that he used to prove P-hardness of the k -WL-equivalence problem. The construction encodes monotone boolean circuits into graphs using different types of gadgets. This simultaneously reduces the monotone circuit value problem, which is known to be hard for polynomial time, to the k -WL-equivalence and the k -WL-identification problem. The main difficulty was to show identification of Grohe's gadgets, specifically his so-called *one-way switches*. We give an alternative construction of these one-way switches based on the CFI-construction. This construction simplifies proofs and more importantly yields graphs with 4-bounded color classes for every k . This shows hardness for the k -WL-equivalence and k -WL-identification problems even for graphs with 4-bounded abelian color classes.

Full proofs of all statements can be found in the full version of this paper [35].

2 The Weisfeiler-Leman Algorithm and Coherent Configurations

Preliminaries. For $n \in \mathbb{N}$, we set $[n] := \{1, \dots, n\}$. For a set A , the set of all k -element subsets of A is denoted by $\binom{A}{k}$. For two runtime-bounding functions f and g with parameters including κ , we write $f \in O_\kappa(g)$ if f/g is bounded by a function of κ . A simple graph is a pair $G = (V(G), E(G))$ of a set $V(G)$ of *vertices* and a set $E(G) \subseteq \binom{V(G)}{2}$ of undirected edges. For a directed graph, we allow $E(G) \subseteq V(G)^2 \setminus \{(v, v) : v \in V(G)\}$. For either graph type, we write uv for the edge $\{u, v\}$ or (u, v) respectively. For a simple or directed graph G , a *vertex-coloring* of G is a map $\chi: V(G) \rightarrow C$ for some finite, ordered set C of colors. Similarly, an *edge-coloring* is a map $\eta: E(G) \rightarrow C$. A (vertex-)color class is a set $\chi^{-1}(c)$ for some vertex color $c \in C$. If all color classes have order at most q , we say that the colored graph (G, χ) has *q -bounded color classes*.

Relational structures are a higher-arity analogue of graphs. Formally, a *k -ary relational structure* \mathfrak{A} is a tuple $(V(\mathfrak{A}), R_1, \dots, R_\ell)$ of vertices $V(\mathfrak{A})$ and relations $R_i \subseteq V(\mathfrak{A})^{r_i}$ with $r_i \leq k$. The number r_i is the *arity* of the relation R_i . We again allow relational structures to come with a vertex-coloring and define q -bounded color classes as before.

An *isomorphism* between graphs G and H is a bijection $\varphi: V(G) \rightarrow V(H)$ such that $uv \in E(G)$ if and only if $\varphi(u)\varphi(v) \in E(H)$. In this case G and H are *isomorphic* and we write $G \cong H$. An isomorphism between edge- or vertex-colored graphs must also preserve the vertex- and edge-colors. Similarly, an isomorphism between (vertex-colored) relational structures is a (color-preserving) bijection between the vertex sets that preserves all relations and their complements. An automorphism is an isomorphism from a structure to itself. We say that a graph or relational structure \mathfrak{A} has *abelian color classes* if for every color class C , the induced substructure $\mathfrak{A}[C]$ has an abelian automorphism group.

Bounded Variable Counting Logics. First-order counting logic C is the extension of first-order logic by the *counting quantifiers* $\exists^{\geq k}$ for all natural numbers k , which state that there exist at least k distinct elements satisfying the formula that follows. But because first-order logic has the ability to simulate the counting quantifier $\exists^{\geq k}$ by a sequence of k usual existential quantifiers, adding counting quantifiers does not actually increase the expressive power of first-order logic. This situation changes when we restrict the number of variables. For a natural number $k \geq 2$, we define *k -variable counting logic* C^k to be the fragment of C which only uses the variables x_1, \dots, x_k . In order to not restrict the expressive power of these logics too much, we do, however, allow *requantifications*, that is, quantifications over a variable within the scope of another quantification over the same variable. As an example, the following is a C^2 -formula

$$\forall x_1 \exists x_2 (Ex_1x_2 \wedge (\exists^{\geq 5} x_1 Ex_2x_1) \wedge \neg \exists^{\geq 6} x_1 Ex_2x_1),$$

which states that every vertex is adjacent to a vertex of degree 5.

A relational structure \mathfrak{A} is *definable* in C^k if there exists some formula $\varphi \in C^k$ which is satisfied by a structure if and only if it is isomorphic to \mathfrak{A} .

The Weisfeiler-Leman Algorithm. The distinguishing power of bounded variable counting logics has another characterization in terms of the Weisfeiler-Leman algorithm. For every $k \geq 2$, the *k -dimensional Weisfeiler-Leman algorithm* (k -WL) computes an isomorphism-invariant coloring of k -tuples of vertices of a given graph G via an iterative refinement process. Initially, the algorithm colors each k -tuple according to its *isomorphism type*, i.e., $\mathbf{x} = (x_1, \dots, x_k), \mathbf{y} = (y_1, \dots, y_k) \in V(G)^k$ get the same color if and only if mapping

$x_i \mapsto y_i$ for every $i \in [k]$ is an isomorphism of the induced subgraphs $G[\{x_1, \dots, x_k\}]$ and $G[\{y_1, \dots, y_k\}]$. In each iteration, this coloring is refined as follows: if $\chi_r^G: V(G)^k \rightarrow C_r$ is the coloring obtained after r refinement rounds, the coloring $\chi_{r+1}^G: V(G)^k \rightarrow C_{r+1}$ is defined as $\chi_{r+1}^G(\mathbf{x}) := (\chi_r^G(\mathbf{x}), M_{\mathbf{x}}^r)$, where

$$M_{\mathbf{x}}^r = \left\{ \left\{ \left(\chi_r^G(\mathbf{x} \frac{y}{1}), \dots, \chi_r^G(\mathbf{x} \frac{y}{k}) \right) : y \in V(G) \right\} \right\}$$

and $\mathbf{x} \frac{y}{i}$ denotes the tuple obtained from \mathbf{x} by replacing the i -th entry by y . If χ_{r+1}^G does not induce a finer color partition on $V(G)^k$ than χ_r^G , then the algorithm terminates and returns the *stable coloring* $\chi_{\infty}^G := \chi_{r+1}^G$. This must happen before the n^k -th refinement round.

We say that k -WL *distinguishes two k -tuples* $\mathbf{x}, \mathbf{y} \in V(G)^k$ if $\chi_{\infty}^G(\mathbf{x}) \neq \chi_{\infty}^G(\mathbf{y})$ and that k -WL *distinguishes two ℓ -tuples* $\mathbf{x}, \mathbf{y} \in V(G)^{\ell}$ for $\ell < k$ if k -WL distinguishes the two k -tuples we get by repeating the last entries of \mathbf{x} respectively \mathbf{y} . In either case, we write $(G, \mathbf{x}) \not\equiv_{k\text{-WL}} (G, \mathbf{y})$. Finally, k -WL *distinguishes two graphs* G and H if the multisets of stable colors computed for the k -tuples of vertices over the two graphs disagree. Otherwise, G and H are k -WL-*equivalent* and we write $G \equiv_{k\text{-WL}} H$. A graph G is *identified* by k -WL if k -WL distinguishes G from every other non-isomorphic graph. Every n -vertex graph is identified by n -WL, and the least number k such that k -WL identifies G is called the *Weisfeiler-Leman dimension* of G , denoted by $\text{WL-dim}(G)$.

k -WL is at least as powerful in distinguishing graphs as $(k-1)$ -WL and this hierarchy does not collapse [11]. Completely analogously, k -WL can be applied to relational structures.

► **Lemma 5** ([11, 26]). *Let \mathfrak{A} and \mathfrak{B} be two relational structures of arity at most k , and $\mathbf{a} \in V(\mathfrak{A})^k$ and $\mathbf{b} \in V(\mathfrak{B})^k$ two tuples of vertices. Then the following are equivalent:*

- (i) *For every C^{k+1} -formula $\varphi(x_1, \dots, x_k)$, we have $(\mathfrak{A}, \mathbf{a}) \models \varphi$ if and only if $(\mathfrak{B}, \mathbf{b}) \models \varphi$, and*
- (ii) *the stable colors computed by k -WL for the tuples \mathbf{a} and \mathbf{b} agree.*

Further, every stable color class is definable by a single C^{k+1} -formula.

In particular, the Weisfeiler-Leman dimension of a structure is precisely one less than the number of variables needed to define the structure in first-order counting logic.

Coherent Configurations. For an introduction to (2-ary) coherent configurations and their connection to the Weisfeiler-Leman algorithm we refer to [15]. For $k \geq 2$, a k -*ary rainbow* is a pair (V, \mathcal{R}) of a finite set of vertices V and a partition \mathcal{R} of V^k , whose elements are called *basis relations*, that satisfies the following two conditions:

- (R1) For every basis relations $R \in \mathcal{R}$, all tuples $\mathbf{x}, \mathbf{y} \in R$ have the same *equality type*, i.e., $x_i = x_j$ if and only if $y_i = y_j$. We also call this the equality type of the relation R .
- (R2) \mathcal{R} is closed under permuting indices: For all basis relations $R \in \mathcal{R}$ and permutations σ of $[k]$, the set $R^{\sigma} := \{(x_{\sigma(1)}, \dots, x_{\sigma(k)}) : (x_1, \dots, x_k) \in R\}$ is a basis relation.

Because the vertex set V is determined by the partition \mathcal{R} , we write \mathcal{R} to denote the rainbow (V, \mathcal{R}) and in this case write $V(\mathcal{R})$ for its vertex set V . A k -*ary coherent configuration* is a k -ary rainbow \mathcal{C} that is stable under k -WL-refinement. Formally, this means that

- (C) for all basis relations $R, R_1, \dots, R_k \in \mathcal{C}$, the *intersection number*

$$p(R; R_1, \dots, R_k) := \left| \left\{ y \in V(\mathcal{C}) : \mathbf{x} \frac{y}{i} \in R_i \text{ for all } i \in [k] \right\} \right|$$

is the same for all choices of $\mathbf{x} \in R$ and is thus well-defined.

For $\ell \leq k$, the partition of k -vertex tuples of an ℓ -ary relational structure according to their isomorphism type always yields a k -ary rainbow. The connection of k -WL and k -ary coherent configurations is that the partition of k -vertex tuples of a graph according to their k -WL-colors always forms a k -ary coherent configuration.

Induced Configurations. If \mathcal{R} is an ℓ -ary rainbow for $\ell \leq k$, we can interpret \mathcal{R} as the k -ary rainbow $\mathcal{R}|^k$ by partitioning k -tuples according to the basis relations of the ℓ -subtuples they contain. Formally, let $\sim_{\mathcal{R}}$ be the equivalence relation on $V(\mathcal{R})^\ell$ whose equivalence classes are the basis relations of \mathcal{R} . We define the equivalence relation $\sim_{\mathcal{R}}^k$ on $V(\mathcal{R})^k$ by writing $\mathbf{x} \sim_{\mathcal{R}}^k \mathbf{y}$ if and only if for all $I \in \binom{[k]}{\ell}$ we have $\mathbf{x}|_I \sim_{\mathcal{R}} \mathbf{y}|_I$, where $\mathbf{x}|_I$ is the subtuple of \mathbf{x} for which all indices not in I are deleted. The basis relations of $\mathcal{R}|^k$ are the equivalence classes of $\sim_{\mathcal{R}}^k$.

For every k -ary rainbow \mathcal{R} , there is a unique coarsest k -ary coherent configuration $\text{WL}_k(\mathcal{R})$ that is at least as fine as \mathcal{R} and is called the *k -ary coherent closure* of \mathcal{R} . For an $\ell \leq k$ and an ℓ -ary rainbow \mathcal{R} , we also write $\text{WL}_k(\mathcal{R})$ for $\text{WL}_k(\mathcal{R}|^k)$. Similarly, for an ℓ -ary relational structure \mathfrak{A} , we write $\text{WL}_k(\mathfrak{A})$ for the partition of $V(\mathfrak{A})^k$ into k -WL-color classes.

Every k -ary coherent configuration \mathcal{C} induces the ℓ -ary coherent configuration $\mathcal{C}|_\ell$ for every $\ell \leq k$ by considering the partition of tuples of the form $(x_1, \dots, x_\ell, \dots, x_\ell) \in V(\mathcal{C})^k$. This ℓ -ary coherent configuration is called the *ℓ -skeleton* of \mathcal{C} . For every basis relation $R \in \mathcal{C}$ and every subset $I \in \binom{[k]}{\ell}$ of the indices, the set $R_I := \{\mathbf{x}|_I : \mathbf{x} \in R\}$ is a basis relation of $\mathcal{C}|_\ell$ and called the *I -face* of R . The 1-skeleton yields a partition of $V(\mathcal{C})$, whose partition classes are called *fibers*. We denote the set of fibers by $F(\mathcal{C})$. \mathcal{C} has *c -bounded fibers* if all fibers of \mathcal{C} have order at most c . If $W \subseteq V(\mathcal{C})$ is a union of fibers, the induced structure $\mathcal{C}[W]$ is again a k -ary coherent configuration. Between two fibers X and Y , the induced configuration $\mathcal{C}|_2$ further induces a partition $\mathcal{C}|_2[X, Y]$ of $X \times Y$, called an *interspace*.

A k -ary coherent configuration \mathcal{C} is *ℓ -induced* if it is the coherent closure of its ℓ -skeleton, i.e., if $\mathcal{C} = \text{WL}_k(\mathcal{C}|_\ell)$. This is equivalent to \mathcal{C} being the coherent closure of some ℓ -ary rainbow. In particular, the k -ary coherent closure of a (directed, colored) graph is 2-induced and the k -ary coherent closure of an ℓ -ary relational structure is ℓ -induced for every $k \geq \ell$.

For a k -ary rainbow $\mathcal{R} = (V, \{R_1, \dots, R_\ell\})$, the vertex-colored k -ary relational structure $(V, R_1, \dots, R_\ell, \chi)$ where χ maps every vertex to its fiber is a *colored variant* of \mathcal{R} . Note that this requires choosing an ordering of the basis relations; colored variants are thus not unique.

Algebraic and Combinatorial Isomorphisms. There are two notions of isomorphism for two k -ary coherent configurations \mathcal{C} and \mathcal{D} . First, a *combinatorial isomorphism* is a bijection $\varphi: V(\mathcal{C}) \rightarrow V(\mathcal{D})$ that preserves the partition into basis relations, i.e., for every basis relation $R \in \mathcal{C}$, the mapped set $R^\varphi := \{(\varphi(x_1), \dots, \varphi(x_k)) : (x_1, \dots, x_k) \in R\}$ is a basis relation of \mathcal{D} . Combinatorial isomorphisms are thus isomorphisms between certain colored variants of \mathcal{C} and \mathcal{D} and the notion also applies to rainbows.

Second, an *algebraic isomorphism* is a map $f: \mathcal{C} \rightarrow \mathcal{D}$ between the two partitions that preserves the intersection numbers. More formally, we require that

- (A1) for all $R \in \mathcal{C}$, the relations R and $f(R)$ have the same equality type,
- (A2) for all $R \in \mathcal{C}$ and permutations σ of $[k]$, we have $f(R^\sigma) = f(R)^\sigma$, and
- (A3) for all $R, T_1, \dots, T_k \in \mathcal{C}$, we have $p(R; T_1, \dots, T_k) = p(f(R); f(T_1), \dots, f(T_k))$,

but Property (A3) already implies the former two. Algebraic isomorphisms can be thought of as maps preserving the Weisfeiler-Leman colors and thus as a functional perspective on Weisfeiler-Leman equivalence. More formally, if for k -ary relational structures \mathfrak{A} and \mathfrak{B} , $f: \text{WL}_k(\mathfrak{A}) \rightarrow \text{WL}_k(\mathfrak{B})$ is an algebraic isomorphism that preserves the relations of \mathfrak{A} and

\mathfrak{B} , then f is the unique map that maps every color class of the stable coloring computed by k -WL on \mathfrak{A} to the corresponding color class of the stable coloring computed by k -WL on \mathfrak{B} . In particular, we get $\mathfrak{A} \equiv_{k\text{-WL}} \mathfrak{B}$ in this case.

If $f: \mathcal{C} \rightarrow \mathcal{D}$ is an algebraic isomorphism, then f induces an algebraic isomorphism $f|_\ell: \mathcal{C}|_\ell \rightarrow \mathcal{D}|_\ell$ for every $\ell \leq k$. If $\mathcal{C} = \text{WL}_k(\mathcal{R})$ for some rainbow \mathcal{R} , f induces a map $f|_{\mathcal{R}}: \mathcal{R} \rightarrow \mathcal{R}^f$ for some rainbow \mathcal{R}^f by sending each basis relation of \mathcal{R} , which is a union of basis relations of \mathcal{C} , to the union of f -images of these basis relations of \mathcal{C} . A combinatorial (respectively algebraic) *automorphism of \mathcal{C}* is a combinatorial (respectively algebraic) isomorphism from \mathcal{C} to itself. Every combinatorial isomorphism induces an algebraic isomorphism, but the converse is not true. Algebraic isomorphisms behave nicely with coherent closures as seen in the next lemma (the proof is analogue to the $k = 2$ case [15, Lemma 2.4]):

- **Lemma 6.** *For all k -ary rainbows \mathcal{R} , algebraic isomorphisms $f: \mathcal{C} \rightarrow \mathcal{D}$, and $\mathcal{C} = \text{WL}_k(\mathcal{R})$*
1. $\mathcal{D} = \text{WL}_k(\mathcal{R}^f)$, in particular, if \mathcal{C} is ℓ -induced, then so is \mathcal{D} ,
 2. f is fully determined by its action on basis relations in \mathcal{R} , and
 3. if $f|_{\mathcal{R}}$ is induced by a combinatorial isomorphism φ , then φ induces f .

A k -ary coherent configuration \mathcal{C} is called *separable* if every algebraic isomorphism $f: \mathcal{C} \rightarrow \mathcal{D}$ from \mathcal{C} is induced by a combinatorial one. There is a close relation to the power of the Weisfeiler-Leman algorithm (the proof is analogue to the $k = 2$ case [15, Theorem 2.5]):

- **Lemma 7.** *Let $\ell \leq k$ and \mathfrak{A} be an ℓ -ary relational structure. Then \mathfrak{A} is identified by the k -dimensional Weisfeiler-Leman algorithm if and only if $\text{WL}_k(\mathfrak{A})$ is separable.*

3 Deciding Identification for Graphs With 5-Bounded Color Classes

As recently shown [15], identification of graphs with 5-bounded color classes by 2-WL is polynomial-time decidable. We extend this result to arbitrary dimensions of the Weisfeiler-Leman algorithm. We adapt the approach of [15] and solve the separability problem for 2-induced k -ary coherent configurations with 5-bounded fibers. We generalize the elimination of interspaces containing a matching and of interspaces of type $2K_{1,2}$: we reduce to *star-free* k -ary coherent configurations. By characterizing separability using certain automorphism groups, we provide a new reduction of the separability problem for such configurations to graph isomorphism for bounded color classes, which can be solved in polynomial time.

Disjoint Unions of Stars. Let \mathcal{C} be a k -ary coherent configuration and $X, Y \in F(\mathcal{C})$ two distinct fibers. A *disjoint union of stars* between X and Y is a basis relation $S \in \mathcal{C}|_2[X, Y]$ such that every vertex in Y is incident to exactly one edge in S . If no interspace of \mathcal{C} contains a disjoint union of stars, then \mathcal{C} is called *star-free*. We show that the separability problem of (2-induced) k -ary coherent configurations reduces to that of star-free ones.

- **Lemma 8.** *Let \mathcal{C} be a k -ary coherent configuration, $X, Y \in F(\mathcal{C})$ two distinct fibers, and $S \in \mathcal{C}|_2[X, Y]$ a disjoint union of stars between X and Y . Then \mathcal{C} is separable if and only if $\mathcal{C} \setminus X := \mathcal{C}[V(\mathcal{C}) \setminus X]$ is separable. Furthermore, if \mathcal{C} is 2-induced, then so is $\mathcal{C} \setminus X$.*

Proof sketch. Let Eq_S be the set of pairs of vertices in Y that have a common S -neighbor in X . We show that the k -ary coherent configuration \mathcal{C} is uniquely determined by the configuration $\mathcal{C} \setminus X$ and the relation Eq_S . For this, consider the function $\nu_S: Y \rightarrow X$ that maps each vertex in Y to its unique neighbor in X . When we apply this map to some of the Y -components of a basis relation $R \in \mathcal{C}$, the resulting set is again a basis relation, and we can obtain every basis relation of $R \in \mathcal{C}$ from basis relations in $\mathcal{C} \setminus X$ in this way.

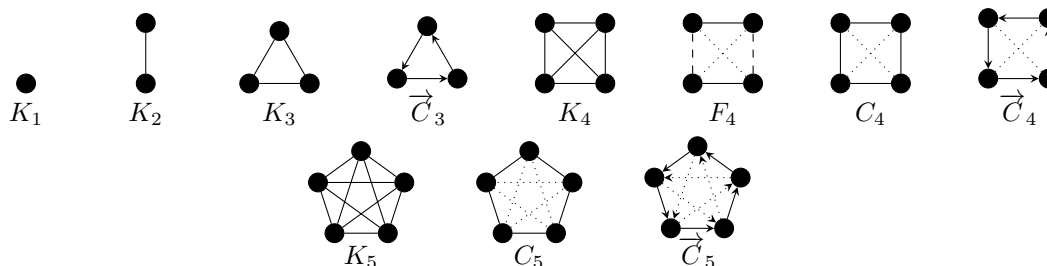


Figure 1 The complete list of 2-ary coherent configurations on a single fiber of order up to 4 from [15], and the three 2-ary coherent configurations on a single fiber of order 5 from [41].

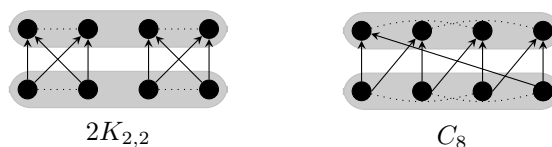


Figure 2 All non-uniform and star-free interspace types between two fibers of order up to 5. In each case, there are at least two basis relations in each fiber, including the drawn matchings, and two basis relations between the fibers: the drawn one and its complement.

We show that both algebraic and combinatorial isomorphisms can detect the uniqueness of this extension in the following sense: every algebraic or combinatorial isomorphism $\mathcal{C} \setminus X \rightarrow \mathcal{D}$ uniquely extends to an algebraic or combinatorial isomorphism $\mathcal{C} \rightarrow \mathcal{D}^*$, for some uniquely determined extension $\mathcal{D}^* \supseteq \mathcal{D}$ by a single fiber. Hence, \mathcal{C} is separable if and only if $\mathcal{C} \setminus X$ is. By analyzing this unique extension, it is moreover clear that it does not affect 2-inducedness. ◀

Lemma 8 allows us to remove fibers that are incident to a disjoint unions of stars without affecting the separability. This simultaneously generalizes the elimination of interspaces containing a matching and the elimination of fibers of size 2 from [15].

5-Bounded Fibers. In order to structurally understand 2-induced k -ary coherent configurations, it mostly suffices to understand their 2-skeletons. The possible isomorphism types of 2-ary coherent configurations on a single fiber of order at most 5 are known [15, 41], see Figure 1. Further, the possible interspaces between fibers of order up to 4 are also known [15]. For fibers $X, Y \in F(\mathcal{C})$, it is always possible that the interspace $\mathcal{C}[X, Y]$ is *uniform*, meaning that it consists of only a single basis relation $X \times Y$. Every interspace between a fiber of size 5 and a fiber of size at most 4 is uniform [15, Lemma 3.1], and every interspace between two fibers of size 5 is either uniform or contains a matching [15, Section 13]. Now, only two possible non-uniform, star-free interspaces remain, which are depicted in Figure 2.

We call an algebraic automorphism f of a k -ary coherent configuration \mathcal{C} *strict* if it fixes every fiber, i.e., it satisfies $f(X) = X$ for every $X \in F(\mathcal{C})$. The strict algebraic automorphisms of \mathcal{C} form a group, which we denote by $\mathbb{A}(\mathcal{C})$. Using the enumeration of fiber and interspace types, we obtain the following reformulation of separability as in [15, Lemma 7.2].

► **Lemma 9.** *A star-free 2-induced k -ary coherent configuration with 5-bounded fibers is separable if and only if every strict algebraic automorphism is induced by a combinatorial one.*

13:10 Computational Complexity of the Weisfeiler-Leman Dimension

Proof sketch. The forward implication is immediate. For the backward implication, consider an algebraic isomorphism $f: \mathcal{C} \rightarrow \mathcal{D}$. Because every coherent configuration of order at most 8 is separable [15], f is induced by a combinatorial isomorphism on every union of two fibers. We pick such a combinatorial isomorphism inducing $f|_2$ for every interspace of type C_8 and everywhere else, we pick combinatorial isomorphisms inducing $f|_2$ just on each fiber. Using the structure of fibers of order at most 5 and their interspaces, we can show that these isomorphisms combine to a combinatorial isomorphism φ inducing $f|_2$ on every fiber. But then, $\varphi^{-1} \circ f|_2$ is a strict algebraic automorphism and is thus induced by a combinatorial automorphism θ . But then, $\varphi \circ \theta$ induces $f|_2$ and thus also f by Lemma 6. ◀

Strict Algebraic Automorphisms. We now sketch a polynomial-time algorithm that decides whether every strict algebraic automorphism of a k -ary coherent configuration is induced by a combinatorial automorphism. We will heavily use that the graph isomorphism problem is polynomial-time solvable for graphs with bounded color classes.

► **Lemma 10** ([6, 16]). *Isomorphism of k -ary relational structures of order n and c -bounded color classes is decidable in time $O_{k,c}(n^{O(k)})$. A generating set of the automorphism group of these structures is computable in time $O_{k,c}(n^{O(k)})$.*

By encoding the algebraic structure of a coherent configuration \mathcal{C} into a relational structure, we obtain the following:

► **Lemma 11.** *There is an algorithm running in time $O_{k,c}(n^{O(k)})$ that, given a k -ary coherent configuration \mathcal{C} of order n with c -bounded fibers, computes a generating set of $\mathbb{A}(\mathcal{C})$.*

Proof. We construct a $(k+1)$ -ary relational structure $\mathfrak{A}_{\mathcal{C}}$ with c^k -bounded color classes such that $\text{Aut}(\mathfrak{A}_{\mathcal{C}}) \cong \mathbb{A}(\mathcal{C})$. The vertices of our constructed structure are the basis relations of \mathcal{C} , and we color each $(k+1)$ -tuple (R, R_1, \dots, R_k) of basis relations using the color $p(R; R_1, \dots, R_k)$. Further, we color each basis relation by the tuple of fibers of its components. Because every fiber is c -bounded, at most c^k basis relations can share a color. Furthermore, it is immediate that the automorphisms of the structure constructed so far naturally correspond to strict algebraic automorphisms of \mathcal{C} . Thus, we can compute a generating set for the group of strict algebraic automorphisms in the required time using Lemma 10. ◀

Similarly, we can reduce the question whether a strict algebraic automorphism is induced by a combinatorial one to the graph isomorphism problem.

► **Lemma 12.** *There is an algorithm running in time $O_{k,c}(n^{O(k)})$ that, given a k -ary coherent configuration \mathcal{C} with c -bounded fibers and a strict algebraic automorphism $f \in \mathbb{A}(\mathcal{C})$, decides whether f is induced by a combinatorial automorphism.*

Proof sketch. Let \mathfrak{C} be an arbitrary colored variant of \mathcal{C} and \mathfrak{C}^f another colored variant such that f is a color-preserving map between the color classes of \mathfrak{C} and \mathfrak{C}^f . Combinatorial automorphisms inducing f correspond to isomorphisms between \mathfrak{C} and \mathfrak{C}^f , meaning that f is induced by a combinatorial automorphism if and only if $\mathfrak{C} \cong \mathfrak{C}^f$. As \mathcal{C} has bounded fibers, so does \mathfrak{C} . Thus, we can decide the latter in the required time by Lemma 10. ◀

► **Corollary 13.** *There is an algorithm running in time $O_{k,c}(n^{O(k)})$ that, given a k -ary coherent configuration \mathcal{C} with c -bounded fibers, decides whether every strict algebraic automorphism of \mathcal{C} is induced by a combinatorial automorphism.*

Proof. Those strict algebraic automorphisms that are induced by combinatorial ones form a subgroup of the group of all strict algebraic automorphisms. Hence, it suffices to compute a generating set of the whole group via Lemma 11 and to decide whether all elements of it are induced by combinatorial automorphisms using Lemma 12. ◀

Finally, we are ready to prove our first theorem.

► **Theorem 1.** *For every k , there is an algorithm that decides the k -WL-identification problem for vertex- and edge-colored, directed graphs with 5-bounded color classes in time $O_k(n^{O(k)})$. If such a graph G is not identified by k -WL, the algorithm provides a witness for this, i.e., a graph H that is not isomorphic to G and not distinguished from G by k -WL.*

Proof. In a first step, we run k -WL on G to get the 2-induced configuration $\mathcal{C} := \text{WL}_k(G)$. By Lemma 7, it remains to decide whether \mathcal{C} is separable. Now, we eliminate disjoint unions of stars using Lemma 8, while maintaining 2-inducedness of \mathcal{C} . By Lemma 9, it remains to decide whether every strict algebraic automorphism is induced by a combinatorial one. This can be achieved using Corollary 13.

If this is the case, the input structure is identified by k -WL. Otherwise, the algorithm actually finds a strict algebraic automorphism f which is not induced by a combinatorial automorphism. By adding back all interspaces containing a disjoint union of stars, we can extend f to an algebraic isomorphism $\hat{f}: \text{WL}_k(G) \rightarrow \mathcal{D}$ which is not induced by a combinatorial isomorphism. But then, we can obtain a witnessing graph H from G by replacing its edge set by its $\hat{f}|_2$ -image and similarly translating vertex- and edge-colors along \hat{f} . ◀

4 Identification for Structures With Bounded Abelian Color Classes

The approach we used in Section 3 to decide the k -WL-identification problem for graphs with 5-bounded color classes does not easily generalize to larger bounds on the color classes or to relational structures of higher arity. In particular, Lemma 9 was crucial in the reduction of k -WL-identification to a statement on certain automorphisms which could be handled using group-theoretic techniques. The proof of the lemma was based on an explicit case distinction on the possible isomorphism types of interspaces, and fails for graphs with larger color classes. In this section, we show that Lemma 9 remains true in the special case of relational structures with bounded abelian color classes, i.e., structures for which the automorphism group of the structure induced on each color class is abelian. Such structures were already considered in the context of descriptive complexity theory [44] and include both CFI-graphs [11] and multipedes [39, 38] over ordered base graphs.

Coherent Configurations With Abelian Fibers. To start, we translate the concept of abelian color classes to the corresponding concept of abelian fibers for k -ary coherent configurations. A combinatorial automorphism φ of a k -ary coherent configuration \mathcal{D} is *color-preserving* if φ fixes every basis relation of \mathcal{D} . This is equivalent to φ being an automorphism of every colored variant of \mathcal{D} or to the algebraic automorphism induced by φ being the identity (recall that combinatorial automorphisms are not required to fix every basis relation, but only the partition of $V(\mathcal{D})^k$ into basis relations). We say that a coherent configuration \mathcal{C} has *abelian fibers* if, for each fiber $X \in F(\mathcal{C})$, the group of color-preserving combinatorial automorphisms of $\mathcal{C}[X]$ is abelian.

► **Lemma 14.** *Let \mathfrak{A} be a relational structure of arity at most k . If \mathfrak{A} has abelian color classes, then $\text{WL}_k(\mathfrak{A})$ has abelian fibers.*

13:12 Computational Complexity of the Weisfeiler-Leman Dimension

We start with a structural lemma, which states that small abelian fibers are always *thin*. For a fiber $X \in F(\mathcal{C})$, a binary basis relation $S \in \mathcal{C}|_2[X]$ is called *thin* if every vertex in X is incident to exactly one ingoing and exactly one outgoing S -edge, that is, if S is either a matching or a union of directed cycles. The fiber X is called *thin* if all basis relations $R \in \mathcal{C}|_2[X]$ are thin and if this is true for all fibers of \mathcal{C} , we say that \mathcal{C} has *thin fibers*.

► **Lemma 15.** *Let \mathcal{C} be a k -ary coherent configuration. Then every abelian fiber of order at most k is thin.*

Proof. Let $X \in F(\mathcal{C})$ be an abelian fiber of order at most k . Then $\mathcal{C}|_2[X]$ is the partition of X^2 into orbits under the natural action of the group of color-preserving automorphisms.

Now, assume that some binary basis relation $S \in \mathcal{C}|_2[X]$ contains two pairs xy and xy' for $x, y, y' \in X$. This implies that there is a color-preserving automorphism φ of $\mathcal{C}[X]$ that maps xy to xy' . But as the group of color-preserving automorphism of $\mathcal{C}[X]$ is abelian and acts transitively on the vertices of X , its point-stabilizers are trivial. Because $\varphi(x) = x$, this implies $\varphi = \text{id}_X$ and thus $y' = \varphi(y) = y$. Thus, the basis relation S is thin. ◀

Next, we need one well-known lemma on the structure of thin fibers, which essentially states that thin fibers correspond to Cayley graphs of their automorphism groups.

► **Lemma 16** ([12, Section 2.1.4]). *Let \mathcal{C} be a 2-ary coherent configuration on a single thin fiber. Then the basis relations of \mathcal{C} are precisely those of the form $S_\varphi := \{x\varphi(x) : x \in V(\mathcal{C})\}$ for color-preserving combinatorial automorphisms φ of \mathcal{C} .*

Separability of Configurations With Bounded Thin Fibers. Next, we show that k -ary coherent configurations with few, thin fibers are separable:

► **Lemma 17.** *Let \mathcal{C} be a k -ary coherent configuration with at most k fibers. If \mathcal{C} has thin fibers, then \mathcal{C} is separable.*

Proof sketch. Let $\mathbf{x} \in V(\mathcal{C})^k$ be a k -tuple of vertices which contains a vertex from every fiber of \mathcal{C} . Then every vertex of \mathcal{C} is the unique outgoing neighbor of some vertex in \mathbf{x} with respect to some thin basis relation. Thus, all vertices of \mathcal{C} are fixed relative to \mathbf{x} .

Now, let \mathfrak{C} be a colored variant of \mathcal{C} . By Lemma 7, \mathcal{C} is separable if and only if \mathfrak{C} is identified by k -WL. We show the latter using the bijective $(k+1)$ -pebble game, which is an Ehrenfeucht-Fraïssé-type game capturing k -WL-equivalence [26]. Indeed, if $\mathfrak{C} \equiv_{k\text{-WL}} \mathfrak{D}$, this corresponds to Duplicator having a winning strategy in this game. But because we can fix every vertex of \mathfrak{C} by only fixing one vertex per color class, we can easily extract an isomorphism $\mathfrak{C} \rightarrow \mathfrak{D}$ from such a winning strategy. ◀

Finally, we are ready to once again reduce the question of separability to only strict algebraic automorphisms, which we can again deal with using Corollary 13.

► **Lemma 18.** *Let \mathcal{C} be a k -ary coherent configurations with thin fibers. Then \mathcal{C} is separable if and only if every strict algebraic automorphism of \mathcal{C} is induced by a combinatorial automorphism.*

Proof sketch. The proof is similar to that of Lemma 9, where instead of using that every 2-ary coherent configuration of order at most 8 is separable, we apply Lemma 17 to get separability of sufficiently small configurations. Afterwards, we use the structure of configurations with thin fibers to show that the local bijections we picked are compatible with the partition in the k -ary interspaces. ◀

► **Theorem 2.** *For every $k \in \mathbb{N}$ and $c, r \leq k$, there is an algorithm that decides the k -WL-identification problem for r -ary relational structures with c -bounded abelian color classes in time $O_k(n^{O(k)})$. If such a structure \mathfrak{A} is not identified by k -WL, the algorithm provides a witness for this, i.e., a second structure \mathfrak{B} that is not isomorphic to \mathfrak{A} and not distinguished from \mathfrak{A} by k -WL.*

Proof. Let \mathfrak{A} be a relational structure of arity r . Then \mathfrak{A} is identified by k -WL if and only if $\text{WL}_k(\mathfrak{A})$ is separable. Because the k -ary coherent configuration $\text{WL}_k(\mathfrak{A})$ has c -bounded thin fibers by Lemmas 14 and 15, Lemma 18 implies that separability of $\text{WL}_k(\mathfrak{A})$ is equivalent to every strict algebraic automorphism of $\text{WL}_k(\mathfrak{A})$ being induced by a combinatorial automorphism. This can be checked in the given time using Corollary 13, and in case of a negative answer, we can construct a non-isomorphic but non-distinguished structure from the strict algebraic automorphism not induced by a combinatorial one as in Theorem 1. ◀

Note that the restriction to relational structures of arity at most k is insubstantial, because the standard variant of the Weisfeiler-Leman algorithm given in Section 2 does not identify any relational structure of arity larger than k , simply because it does not consider tuples of length larger than k and thus cannot even detect whether a relation of arity larger than k is empty. While there are variants of k -WL which identify some $(k+1)$ -ary relational structures, these variants can be treated similarly to decide identification by those algorithms.

5 Hardness

We now prove hardness results that complement the positive results in the previous two sections. In the case that the dimension k is part of the input, the k -WL-equivalence problem and the k -WL-identification problem are co-NP-hard and NP-hard, respectively. We use that deciding whether a cubic graph has tree-width k is NP-hard [9]. The CFI-construction [11] assigns to a graph G two CFI-graphs, which are distinguished by k -WL if and only if G has tree-width at most k [8, 22]. If G is cubic, then the CFI-graphs are polynomial-time computable and thus we reduced to k -WL-equivalence. To show hardness of k -WL-identification, we show that the CFI-graphs are actually identified by k -WL if the tree-width of G is at most k . Hardness of the k -WL-equivalence problem was independently observed by Seppelt [42].

► **Theorem 3.** *The problem of deciding, given a graph G and a natural number k , whether the Weisfeiler-Leman dimension of G is at most k is NP-hard, both over uncolored simple graphs, and over simple graphs with 4-bounded color classes.*

► **Theorem 19.** *The problem of deciding, for a given pair of graphs G and H and a natural number $k \geq 1$, whether $G \equiv_{k\text{-WL}} H$ is co-NP-hard, both over uncolored simple graphs, and over simple graphs with 4-bounded abelian color classes.*

P-Hardness for Fixed Dimension. We again turn to the k -WL-identification problem for a fixed dimension $k \geq 2$, and show that both over uncolored simple graphs and over simple graphs with 4-bounded abelian color classes, the problem is P-hard under logspace-uniform AC_0 -reductions. We reduce from the P-hard monotone circuit value problem MCVP [17]. Our construction of a graph from a monotone circuit closely resembles the reductions of Grohe [19] to show P-hardness of the k -WL-equivalence problem. A similar reduction was also used to prove P-hardness of the identification problem for the color refinement algorithm (1-WL) [2].

The reduction is based on so-called *one-way switches*, which were introduced by Grohe [19]. These graph gadgets allow color information computed by the Weisfeiler-Leman algorithm to pass in one direction, but block it from passing in the other. And while Grohe provides one-way switches for every dimension of the Weisfeiler-Leman algorithm, his gadgets have large color classes and are difficult to analyze. Instead, we give a new construction of such gadgets with 4-bounded color classes. We then use these one-way switches to construct a graph from an instance of the monotone circuit value problem from the identification of which we can read off the answer to the initial MCVP-query.

One-Way Switches. Fix a dimension $k \geq 2$ of the Weisfeiler-Leman algorithm. A *k-one-way switch* is a graph gadget with a pair of *input vertices* $\{y_1, y_2\}$ and a pair of *output vertices* $\{x_1, x_2\}$, which each form a color class of size 2. A pair of vertices is *split* if the two vertices are colored differently and *k-WL splits* a pair if the coloring computed by *k-WL* splits the pair. The crucial property of a *k-one-way switch* is the following: if the input pair $\{y_1, y_2\}$ of the one-way switch is split, then *k-WL* also splits the output pair, but not the other way around. One-way switches thus only allow one-way flow of *k-WL*-color information. In contrast to Grohe's gadgets, our one-way switches are based on the CFI-construction [11]. We give only a brief sketch of the properties of our one-way switches here, and postpone their precise construction and properties to Appendix A.

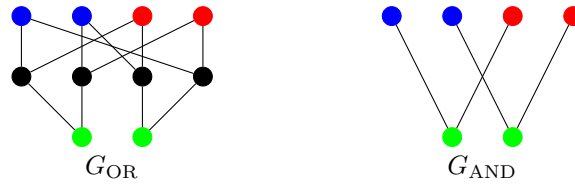
► **Lemma 20 (simplified).** *For every $k \geq 2$, there is a colored graph O^k with 4-bounded abelian color classes, called *k-one-way switch*, with an input pair $\{y_1, y_2\}$ and an output pair $\{x_1, x_2\}$, neither of which is split by *k-WL*, such that*

1. *the graph O_{split}^k obtained by splitting the input pair $\{y_1, y_2\}$ is identified by *k-WL*,*
2. **k-WL* splits the output pair $\{x_1, x_2\}$ of O_{split}^k , and*
3. *if we split the output pair $\{x_1, x_2\}$, *k-WL* still does not split the input pair $\{y_1, y_2\}$.*

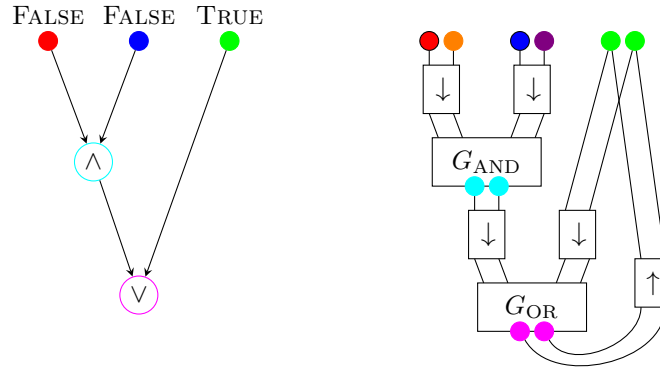
From Monotone Circuits to Graphs. We reduce the monotone circuit value problem to the *k-WL*-identification problem. A monotone circuit M is a circuit consisting of input nodes with values TRUE or FALSE, inner nodes, which are either AND- or OR-nodes with two inputs each, and a distinguished output node. We write $V(M)$ for the set of nodes of M . With a monotone circuit M , we associate the evaluation function $\text{val}_M: V(M) \rightarrow \{\text{TRUE}, \text{FALSE}\}$, which is defined in the expected way. The monotone circuit value problem asks whether the output node of a given monotone circuit evaluates to TRUE and is P-hard by [17].

Let M be a monotone circuit. We construct a colored graph G_M such that for every node $a \in V(M)$, there is a vertex pair $\{a_1, a_2\}$ in G_M that will be split by *k-WL* if and only if $\text{val}_M(a) = \text{FALSE}$. Up to the construction of the one-way switches, the construction of G_M is similar to constructions employed in [19] and [2]. We use two graphs G_{OR} and G_{AND} (see Figure 3) as gadgets to simulate logic gates. These gadgets both have two input pairs and one output pair such that exchanging the two output vertices by an automorphism requires the two vertices of one (for G_{OR}) or both (for G_{AND}) input pairs to also be exchanged.

The formal construction of G_M is depicted in Figure 4. For every node a of M , we add a pair of vertices $\{a_1, a_2\}$ forming a color class of G_M . To encode the input values of the circuit, we split every pair $\{a_1, a_2\}$ corresponding to an input node a of value FALSE. For every AND-node $a \in V(M)$ with input nodes b and b' , we add a freshly colored copy of the gadget G_{AND} from Figure 3 and identify its output pair with the pair $\{a_1, a_2\}$. Next, we connect its two input pairs via freshly colored one-way switches O_{ba}^k and $O_{b'a}^k$ to the pairs $\{b_1, b_2\}$ and $\{b'_1, b'_2\}$ respectively. More precisely, we identify the input pairs of these



■ **Figure 3** The gadgets G_{OR} and G_{AND} encoding OR- and AND-gates respectively. The two vertex pairs at the top are their *input pairs* and the bottom pair is their *output pair*.



■ **Figure 4** A simple monotone circuit M and the graph G_M obtained from it.

one-way switches with $\{b_1, b_2\}$ or $\{b'_1, b'_2\}$ respectively and identify their output pair with the respective input pair of the copy is identif of G_{AND} . Analogously, we add a copy of G_{OR} for every OR-node $a \in V(M)$ and connect it to its input via one-way switches as before.

This concludes the translation of the circuit itself, but for our reduction to the identification problem we need one more step: we connect all input pairs $\{a_1, a_2\}$ with $\text{val}_M(a) = \text{TRUE}$ to the output pair $\{c_1, c_2\}$ via additional one-way switches O_{ca}^k , whose input pair we identify with $\{c_1, c_2\}$ and whose output pair we identify with $\{a_1, a_2\}$. Let G_M be the resulting graph. Because we color different gadgets using distinct colors, and every gadget has 4-bounded color classes, the resulting graph G_M also has 4-bounded color classes and indeed, these color classes could also be made abelian by introducing colored edges within the gadgets. The following lemma is proven similar to [19, Section 5.4] and [2, Theorem 7.11] by using the properties of the one-way switches to bound the distinguishing power of k -WL on G_M in terms of its distinguishing power on the individual gadgets.

Recall that G_M contains, for every node a of M , a vertex pair $\{a_1, a_2\}$. The essential property of the encoding of monotone circuits as graphs is the following:

► **Lemma 21.** *For every monotone circuit M with output node c and every node a of M , we have $(G_M, a_1) \equiv_{k\text{-WL}} (G_M, a_2)$ if and only if $\text{val}_M(a) = \text{val}_M(c) = \text{TRUE}$.*

► **Corollary 22.** *The k -WL-equivalence problem for vertices is P -hard under uniform AC_0 -reductions, both over simple graphs with 4-bounded abelian color classes, and over uncolored simple graphs.*

Consider now the modified graph G_M^* that we get by adding another freshly colored one-way switch O_*^k whose input pair is $\{c_1, c_2\}$, i.e., the vertex pair corresponding to the output node of the circuit M . Furthermore, we split the output pair of O_*^k . Note that when splitting the

output pair, we can choose which of the two vertices to give a fresh color to. We show that these two choices lead to non-isomorphic graphs which are distinguished by k -WL if and only if the circuit evaluates to FALSE.

► **Lemma 23.** *For every monotone circuit M , the graph G_M^* is identified by k -WL if and only if $\text{val}_M(c) = \text{FALSE}$.*

Proof sketch. If $\text{val}_M(c) = \text{FALSE}$, then all input and output pairs in G_M^* are split. Because all gadgets in G_M^* are identified when their input and output pairs are split, and different gadgets only interact at these split pairs, the whole graph G_M^* is identified.

Conversely, assume $\text{val}_M(c) = \text{TRUE}$, and let $(G_M^*)'$ be the graph constructed just like G_M^* , but with the colors of the two output vertices of the one-way switch O_*^k exchanged. Because the pair $\{c_1, c_2\}$ is not split in G_M , k -WL cannot distinguish the two output vertices of O_*^k , which means that it cannot distinguish the non-isomorphic graphs G_M^* and $(G_M^*)'$. ◀

► **Theorem 4.** *For every $k \geq 1$, the k -WL-identification problem is P-hard under uniform AC_0 -reductions over both uncolored simple graphs, and simple graphs with 4-bounded abelian color classes.*

6 Conclusion

We have shown on the one hand that when the dimension k is part of the input, the k -WL-equivalence problem and the k -WL-identification problem are co-NP-hard and NP-hard, respectively.

On the other hand, when the dimension k is fixed, the equivalence problem is trivially solvable in polynomial time, and we have shown that the identification problem is solvable in polynomial time over graphs with 5-bounded color classes and on relational structures with k -bounded abelian color classes. Still, the identification problem is P-hard in both cases. As an immediate corollary, we obtain the same polynomial-time solvability and hardness results for definability and equivalence in the bounded-variable logic with counting C^k .

It would be interesting to know whether the k -WL-identification problem can be solved in polynomial time for larger color classes or indeed on general graphs when k is fixed. Indeed, our NP-hardness reduction was based on whether the tree-width of a given graph is at most k , which can be solved in linear time for every fixed k [7], and thus does not even yield a super-linear lower bound when k is fixed. Still, we would expect that neither the identification nor the equivalence problem can be solved in time $n^{o(k)}$. It might be fruitful to study these problems from the lens of parameterized complexity or provide lower complexity bounds based on the (strong) exponential time hypothesis.

References

- 1 Markus Anders and Pascal Schweitzer. Parallel computation of combinatorial symmetries. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPICs*, pages 6:1–6:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ESA.2021.6.
- 2 Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. Graph isomorphism, color refinement, and compactness. *Comput. Complex.*, 26(3):627–685, 2017. doi:10.1007/S00037-016-0147-6.
- 3 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.

- 4 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *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. doi:10.1145/2897518.2897542.
- 5 László Babai and Ludek Kucera. Canonical labelling of graphs in linear average time. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 39–46. IEEE Computer Society, 1979. doi:10.1109/SFCS.1979.8.
- 6 László Babai. Monte-Carlo algorithms in graph isomorphism testing. Technical Report 79-10, Université de Montréal, 1979.
- 7 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.
- 8 Hans L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 9 Hans L. Bodlaender, Édouard Bonnet, Lars Jaffke, Dušan Knop, Paloma T. Lima, Martin Milanič, Sebastian Ordyniak, Sukanya Pandey, and Ondřej Suchý. Treewidth is NP-complete on cubic graphs. In *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPICs*, pages 7:1–7:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.IPEC.2023.7.
- 10 Béla Bollobás. Distinguishing vertices of random graphs. *North-holland Mathematics Studies*, 62:33–49, 1982. doi:10.1016/S0304-0208(08)73545-X.
- 11 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. doi:10.1007/BF01305232.
- 12 G. Chen and I. Ponomarenko. *Lectures on Coherent Configurations*. Central China Normal University Press, 2019. A draft is available at <https://www.pdmi.ras.ru/~inp/>.
- 13 Anuj Dawar and David Richerby. The power of counting logics on restricted classes of finite structures. In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 84–98. Springer, 2007. doi:10.1007/978-3-540-74915-8_10.
- 14 Zdenek Dvorák. On recognizing graphs by numbers of homomorphisms. *J. Graph Theory*, 64(4):330–342, 2010. doi:10.1002/JGT.20461.
- 15 Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. Identifiability of graphs with small color classes by the Weisfeiler-Leman algorithm. *SIAM J. Discret. Math.*, 35(3):1792–1853, 2021. doi:10.1137/20M1327550.
- 16 Merrick Furst, John Hopcroft, and Eugene M. Luks. A subexponential algorithm for trivalent graph isomorphism. Technical report, Cornell University, USA, 1980.
- 17 Leslie M. Goldschlager. The monotone and planar circuit value problems are log space complete for P. *SIGACT News*, 9(2):25–29, 1977. doi:10.1145/1008354.1008356.
- 18 Erich Grädel and Wied Pakusa. Rank logic is dead, long live rank logic! *J. Symb. Log.*, 84(1):54–87, 2019. doi:10.1017/jsl.2018.33.
- 19 Martin Grohe. Equivalence in finite-variable logics is complete for polynomial time. *Comb.*, 19(4):507–532, 1999. doi:10.1007/S004939970004.
- 20 Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *J. ACM*, 59(5):27:1–27:64, 2012. doi:10.1145/2371656.2371662.
- 21 Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi:10.1017/9781139028868.
- 22 Martin Grohe, Moritz Lichter, Daniel Neuen, and Pascal Schweitzer. Compressing CFI graphs and lower bounds for the Weisfeiler-Leman refinements. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 798–809. IEEE, 2023. doi:10.1109/FOCS57990.2023.00052.

- 23 Martin Grohe and Julian Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer, 1999. doi:10.1007/3-540-49257-7_6.
- 24 Martin Grohe and Daniel Neuen. Canonisation and definability for graphs of bounded rank width. *ACM Trans. Comput. Log.*, 24(1):6:1–6:31, 2023. doi:10.1145/3568025.
- 25 Martin Grohe and Martin Otto. Pebble games and linear equations. *J. Symb. Log.*, 80(3):797–844, 2015. doi:10.1017/JSL.2015.28.
- 26 Lauri Hella. Logical hierarchies in PTIME. *Inf. Comput.*, 129(1):1–19, 1996. doi:10.1006/INCO.1996.0070.
- 27 Neil Immerman and Eric S. Lander. *Describing Graphs: A First-Order Approach to Graph Canonization*, pages 59–81. Springer New York, New York, NY, 1990. doi:10.1007/978-1-4612-4478-3_5.
- 28 Tommi A. Junttila and Petteri Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the Nine Workshop on Algorithm Engineering and Experiments, ALENEX 2007, New Orleans, Louisiana, USA, January 6, 2007*. SIAM, 2007. doi:10.1137/1.9781611972870.13.
- 29 Tommi A. Junttila and Petteri Kaski. Conflict propagation and component recursion for canonical labeling. In *Theory and Practice of Algorithms in (Computer) Systems - First International ICST Conference, TAPAS 2011, Rome, Italy, April 18-20, 2011. Proceedings*, volume 6595 of *Lecture Notes in Computer Science*, pages 151–162. Springer, 2011. doi:10.1007/978-3-642-19754-3_16.
- 30 Sandra Kiefer, Iliia Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. *J. ACM*, 66(6):44:1–44:31, 2019. doi:10.1145/3333003.
- 31 Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs identified by logics with counting. *ACM Trans. Comput. Log.*, 23(1):1:1–1:31, 2022. doi:10.1145/3417515.
- 32 Ludek Kucera. Canonical labeling of regular graphs in linear average time. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 271–279. IEEE Computer Society, 1987. doi:10.1109/SFCS.1987.11.
- 33 Moritz Lichter. Separating rank logic from polynomial time. *J. ACM*, 70(2), March 2023. doi:10.1145/3572918.
- 34 Moritz Lichter. Witnessed symmetric choice and interpretations in fixed-point logic with counting. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *LIPICs*, pages 133:1–133:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.ICALP.2023.133.
- 35 Moritz Lichter, Simon Raßmann, and Pascal Schweitzer. Computational complexity of the Weisfeiler-Leman dimension. *CoRR*, abs/2402.11531, 2024. doi:10.48550/arXiv.2402.11531.
- 36 Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014. doi:10.1016/J.JSC.2013.09.003.
- 37 Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4602–4609. AAAI Press, 2019. doi:10.1609/AAAI.V33I01.33014602.
- 38 Daniel Neuen and Pascal Schweitzer. Benchmark graphs for practical graph isomorphism. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, volume 87 of *LIPICs*, pages 60:1–60:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.ESA.2017.60.
- 39 Daniel Neuen and Pascal Schweitzer. An exponential lower bound for individualization-refinement algorithms for graph isomorphism. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 138–150. ACM, 2018. doi:10.1145/3188745.3188900.

- 40 Thomas Schneider and Pascal Schweitzer. An upper bound on the Weisfeiler-Leman dimension, 2024. arXiv:2403.12581, doi:10.48550/arXiv.2403.12581.
- 41 Kyoungah See and Sung Y. Song. Association schemes of small order. *Journal of Statistical Planning and Inference*, 73(1):225–271, 1998. doi:10.1016/S0378-3758(98)00064-0.
- 42 Tim Seppelt. An Algorithmic Meta Theorem for Homomorphism Indistinguishability. In *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:19, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2024.82.
- 43 B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsia, Seriya 2*, 9:12–16, 1968. An english translation due to Grigory Ryabov is available at https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf.
- 44 Faried Abu Zaid, Erich Grädel, Martin Grohe, and Wied Pakusa. Choiceless polynomial time on structures with small abelian colour classes. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 50–62. Springer, 2014. doi:10.1007/978-3-662-44522-8_5.

A Construction of One-Way Switches

We construct our one-way switches based on the CFI-construction, and the proof of the properties heavily uses the bijective pebble game. We thus start with a short introduction to these.

The Bijective Pebble Game. The question whether C^{k+1} or k -WL can distinguish structures \mathfrak{A} and \mathfrak{B} has another characterization in terms of the so-called *bijective $(k+1)$ -pebble game*. In this game, there are two players: Spoiler and Duplicator. Game positions are partial maps $\mathbf{g} \mapsto \mathbf{h}$ between G and H , where both tuples contain at most $k+1$ elements. We also sometimes identify such partial maps with the set $P = \{g_i \mapsto h_i : i \leq |\mathbf{g}|\}$.

We think of these maps as $k+1$ pairs of corresponding pebbles placed in the two graphs. If such a partial map is not a partial isomorphism, i.e., not an isomorphisms on the induced subgraphs, Spoiler wins immediately.

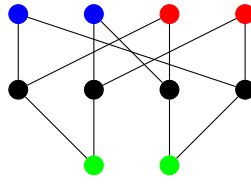
Otherwise, at the beginning of each turn, Spoiler picks up one pebble pair, either from the board if all $k+1$ pairs are placed, or from the side if there are pebble pairs left. Duplicator responds by giving a bijection $\varphi: V(G) \rightarrow V(H)$ between the two graphs. Spoiler then places the pebble pair they picked up on a pair $(g, \varphi(g))$ of vertices of their choice. The game then continues in the resulting new position.

We say that Spoiler wins if the graphs have differing cardinality or they can reach a position that is no longer a partial isomorphism (and thus win immediately). Duplicator wins the game if Duplicator can find responses to Spoiler’s moves indefinitely.

Lemma 5 now has the following extension:

► **Lemma 24** ([11], [26]). *Let \mathfrak{A} and \mathfrak{B} be two relational structures of arity at most k , and $\mathbf{a} \in V(\mathfrak{A})^k$ and $\mathbf{b} \in V(\mathfrak{B})^k$ two tuples of vertices. Then the following are equivalent:*

- (i) *Duplicator has a winning strategy in position $\mathbf{a} \mapsto \mathbf{b}$ of the bijective $(k+1)$ -pebble game between \mathfrak{A} and \mathfrak{B} ,*
- (ii) *for every C^{k+1} -formula $\varphi(x_1, \dots, x_k)$, we have $(G, \mathbf{g}) \models \varphi$ if and only if $(H, \mathbf{h}) \models \varphi$,*
- (iii) *the stable colors computed by k -WL for the tuples \mathbf{a} and \mathbf{b} agree.*



■ **Figure 5** A CFI-gadget for a vertex of degree 3, consisting of four inner vertices and three outer pairs.

The CFI-Construction. CFI-graphs are certain graphs with high Weisfeiler-Leman dimension [11]. To construct them, we start with a *base graph* G , which is a connected simple graph, and a function $f: E(G) \rightarrow \mathbb{F}_2$. For a vertex $v \in V(G)$, we denote the set of edges incident to v by $E[v] := \{uv: v \in N_G(v)\} \subseteq E(G)$. Now, to construct the CFI-graph $\text{CFI}(G, f)$, we replace each vertex $v \in V(G)$ by a gadget X_v which consists of *inner vertices* $I_v := \{v\} \times \{\mathbf{x} \in \mathbb{F}_2^{E[v]}: \sum \mathbf{x} = 0\}$ and *outer vertices* $\{v\} \times \{(e, i): e \in E[v], i \in \mathbb{F}_2\}$. Inside each gadget, the inner and outer vertices each form an independent set, and an inner vertex (v, \mathbf{x}) and outer vertex (v, e, i) are connected by an edge if and only if $\mathbf{x}_e = i$. The resulting gadget for a vertex of degree 3 is depicted in Figure 5.

Next, we define the edge set between different gadgets. For every edge $e = uv \in E(G)$, we connect the outer vertices (u, e, i) and (v, e, j) if and only if $i + j = f(e)$ and add no further edges. Thus, corresponding outer vertex pairs (u, e, \cdot) and (v, e, \cdot) are always connected by a matching, which is either *untwisted* if $f(e) = 0$, or *twisted* if $f(e) = 1$.

Finally, we define a vertex coloring on this graph. For every vertex v , we turn the set I_v of inner vertices into a color class of size $2^{d(v)-1}$. Moreover, we turn each outer pair $\{(v, e, 0), (v, e, 1)\}$ into a color class of size 2. This finishes the construction of CFI-graphs.

It turns out that for two functions $f, g: E(G) \rightarrow \mathbb{F}_2$, we have $\text{CFI}(G, f) \cong \text{CFI}(G, g)$ if and only if $\sum f = \sum g$, meaning that every even number of twists cancels out. Thus, we also write $\text{CFI}(G, 0)$ and $\text{CFI}(G, 1)$ for the *untwisted* and *twisted* CFI-graphs over the base graph G .

To understand the power of the Weisfeiler-Leman algorithm on CFI-graphs, it is convenient to study *tree-width*, which is a graph parameter that intuitively measures how far a graph is from being a tree. In this work, we do not need the formal definition of tree-width, and refer to [8]. The power of the Weisfeiler-Leman algorithm to distinguish CFI-graphs can now conveniently be expressed in terms of the tree-width of the base graphs, see [22].

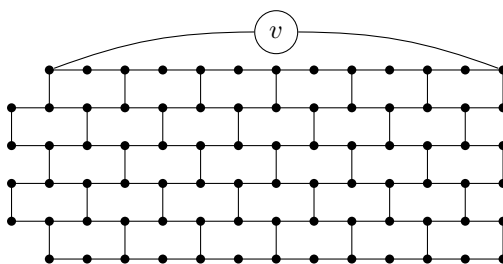
► **Lemma 25.** *For every base graph G of tree-width $\text{tw}(G) \geq 2$, we have*

$$\text{WL-dim}(\text{CFI}(G, 0)) = \text{WL-dim}(\text{CFI}(G, 1)) = \text{tw}(G).$$

Construction of the One-Way Switches. We start by defining a base graph. Consider a wall graph consisting of $k - 1$ rows of k bricks each. Then, we attach a new vertex v to the two upper corner vertices of the first row. The resulting graph B_k is depicted in Figure 6.

► **Lemma 26.** *The graph B_k has tree-width $k + 1$, while $B_k - v$ has tree-width k .*

Now, we are ready to construct our one-way switches. In our proofs, we actually need a more explicit version of Lemma 20 in order to precisely control the expressive power for the Weisfeiler-Leman algorithm on the whole graph in terms of its expressive power on the individual gadgets:



■ **Figure 6** The base graph B_6 of the CFI-graphs underlying our one-way switches.

► **Lemma 27** (compare [19, Lemma 14]). *For every $k \geq 2$, there is a colored graph O^k with 4-bounded color classes, called k -one-way switch, with an input pair $\{y_1, y_2\}$ and an output pair $\{x_1, x_2\}$ satisfying the following properties:*

1. *The graph O_{split}^k obtained by splitting the input pair $\{y_1, y_2\}$ is identified by k -WL.*
 2. *k -WL splits the output pair $\{x_1, x_2\}$ of O_{split}^k .*
 3. *There is no automorphism of O^k exchanging the output vertices x_1 and x_2 .*
- Furthermore, there are sets of positions in the bijective $(k + 1)$ -pebble game between O^k and itself, called trapped and twisted such that*
4. *every trapped or twisted position is a partial isomorphism,*
 5. *Duplicator can avoid non-trapped positions from trapped ones and non-twisted positions from twisted ones,*
 6. *for every trapped position $\mathbf{a} \mapsto \mathbf{b}$, the position $\mathbf{a}x_1 \mapsto \mathbf{b}x_1$ is also trapped,¹*
 7. *for every twisted position $\mathbf{a} \mapsto \mathbf{b}$, the position $\mathbf{a}x_1 \mapsto \mathbf{b}x_2$ is also twisted.*
 8. *the positions $y_1y_2 \mapsto y_1y_2$ and $y_1y_2 \mapsto y_2y_1$ are both trapped and twisted,*
 9. *every subposition of a trapped position is trapped, and every subposition of a twisted position is twisted*

Proof. Let O^k be the (untwisted) CFI-graph of B_k , but with a CFI-gadget of degree 3 added for the vertex v instead of a gadget of degree 2. This leaves one outer pair of this gadget free which we use as our output pair $\{x_1, x_2\}$. Furthermore, we use one of the other two outer pairs of this same CFI-gadget as the input pair $\{y_1, y_2\}$.

Now, if we fix the output pair $\{x_1, x_2\}$ by individualizing one of the two vertices, the resulting graph corresponds to the usual CFI-graph of H , while switching the pair $\{x_1, x_2\}$ corresponds to the twisted CFI-graph of H . In particular, as these graphs are not isomorphic, there is no automorphism of O^k switching the pair $\{x_1, x_2\}$, which proves Property 3.

Moreover, splitting the input pair $\{y_1, y_2\}$ has the same effect to the power of k -WL as removing one of the two edges incident to v in the base graph B_k has. When removing this edge in the base graph, the resulting graph is essentially equivalent to the CFI-graph of the $k \times (k + 1)$ -wall graph with one corner vertex replaced by a CFI-gadget of degree 3 instead of 2. Because exchanging the two vertices of the free outer pair of this degree-3 gadget interchanges the twisted and untwisted CFI-graphs over the base graph, and k -WL can distinguish CFI-graphs from all other graphs, the resulting graph is identified by k -WL. This proves Property 1.

¹ If the position $\mathbf{a}x_1 \mapsto \mathbf{b}x_1$ contains more than $k + 1$ pebbles, this means that every subposition on at most $k + 1$ pebbles is trapped.

13:22 Computational Complexity of the Weisfeiler-Leman Dimension

To show Property 2, we start the bijective $(k + 1)$ -pebble game in position $x_1 \mapsto x_2$. Then, Spoiler uses the usual strategy of pebbling a wall which they then move from one side of the wall graph to the other. But because the game started in position $x \mapsto x'$, the two graphs the game is played on differ in a twist which will finally force Duplicator to lose.

Now, consider again the original graph O^k without splitting the input pair. On this graph, we can extend every winning position for Duplicator in the bijective k -pebble game between the untwisted CFI-graph $\text{CFI}(B_k, 0)$ and the twisted CFI-graph $\text{CFI}(B_k, 1)$ to a position in the bijective k -pebble game between O^k and itself which is compatible with $x_1 \mapsto x_2$. Similarly, we can extend every winning position for Duplicator in the bijective k -pebble game between the untwisted CFI-graph $\text{CFI}(B_k, 0)$ and itself to a position between O^k and itself which is compatible with $x_1 \mapsto x_1$.

We call the former positions *twisted* and the latter positions *trapped*. Properties 4, 6, 7 and 9 are then immediate, and Property 5 follows from Lemma 25 together with Lemma 26.

Because v lies on a cycle in B_k , there exists an automorphism of $\text{CFI}(B_k)$ which twists both outer pairs of the gadget corresponding to v . Lifting this automorphism to O^k yields an automorphism switching y_1 and y_2 whilst fixing x_1 and x_2 . This proves Property 8. ◀