# Symmetric-Difference (Degeneracy) and Signed Tree Models


**Édouard Bonnet**
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

**Julien Duron**
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

**John Sylvester**
Department of Computer Science, University of Liverpool, UK

**Viktor Zamaraev**
Department of Computer Science, University of Liverpool, UK



## Abstract

We introduce a dense counterpart of graph degeneracy, which extends the recently-proposed invariant symmetric difference. We say that a graph has *sd-degeneracy* (for symmetric-difference degeneracy) at most $d$ if it admits an elimination order of its vertices where a vertex $u$ can be removed whenever it has a $d$-twin, i.e., another vertex $v$ such that at most $d$ vertices outside $\{u, v\}$ are neighbors of exactly one of $u, v$. The family of graph classes of bounded sd-degeneracy is a superset of that of graph classes of bounded degeneracy or of bounded flip-width, and more generally, of bounded symmetric difference. Unlike most graph parameters, sd-degeneracy is not hereditary: it may be strictly smaller on a graph than on some of its induced subgraphs. In particular, every $n$-vertex graph is an induced subgraph of some $O(n^2)$-vertex graph of sd-degeneracy 1. In spite of this and the breadth of classes of bounded sd-degeneracy, we devise $\tilde{O}(\sqrt{n})$-bit adjacency labeling schemes for them, which are optimal up to the hidden polylogarithmic factor. This is attained on some even more general classes, consisting of graphs $G$ whose vertices bijectively map to the leaves of a tree $T$, where transversal edges and anti-edges added to $T$ define the edge set of $G$. We call such graph representations *signed tree models* as they extend the so-called tree models (or twin-decompositions) developed in the context of twin-width, by adding transversal anti-edges.

While computing the degeneracy of a graph takes linear time, we show that determining its symmetric difference is para-co-NP-complete. This may seem surprising as symmetric difference can serve as a short-sighted first approximation of twin-width, whose computation is para-NP-complete. Indeed, we show that deciding if the symmetric difference of an input graph is at most 8 is co-NP-complete. We also show that deciding if the sd-degeneracy is at most 6 is NP-complete, contrasting with the symmetric difference.



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

**Keywords and phrases** symmetric difference, degeneracy, adjacency labeling schemes, NP-hardness

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2024.32

**Related Version** *Full Version*: https://arxiv.org/abs/2405.09011 [23]


## 1 Introduction

There are two theories of sparse graphs: the so-called Sparsity Theory pioneered by Nešetřil and Ossona de Mendez [18], and the theory behind the equivalent notions of bounded degeneracy, maximum average degree, subgraph density, and arboricity. One of the many merits of the former theory is to capture efficient first-order model checking within subgraph-closed classes, with the so-called *nowhere dense* classes [12]. *Monadic stability* constitutes a dense analogue of nowhere denseness with similar algorithmic properties [10]. The second



49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 32; pp. 32:1–32:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

theory has, as we will see, simple but useful connections with the chromatic number and adjacency labeling schemes. One of our two main motivations is to introduce and explore dense analogues of it.

The degeneracy of a graph $G$ is the minimum integer $d$ such that every induced subgraph of $G$ has a vertex of degree at most $d$. As removing vertices may only decrease the degree of the remaining ones, checking that the degeneracy is at most $d$ can be done greedily. This prompts the following equivalent definition, equal to the *coloring number*[1] minus one [11, 9]. A graph has degeneracy at most $d$ if there is a total order, called *degeneracy ordering*, on its vertices such that every vertex $v$ has at most $d$ neighbors following $v$ in the order. The degeneracy is then the least integer $d$ such that an ordering witnessing degeneracy at most $d$ exists. Given such an ordering, a graph can be properly $(d+1)$-colored by a greedy strategy: use the smallest available color looping through vertices in the given order. Another advantage of the definition via degeneracy ordering is that it yields a polynomial-time algorithm to compute the degeneracy. While the graph is nonempty, find a vertex of minimum degree, append it to the order, and remove it from the graph. Degeneracy is frequently used to bound the chromatic number from above. For instance, until recently [19] the Kostochka–Thomason degeneracy bound of graphs without $K_t$ minor [15, 20] was the best way we knew of coloring these graphs.

Another application of bounding the degeneracy is to obtain implicit representations. Indeed graphs of bounded degeneracy admit $f(n)$-bit *adjacency labeling schemes* with $f(n) = O(\log n)$.[2] In other words, given a class of graphs of degeneracy at most $d$, there exists an algorithm, called *decoder*, such that the vertices of any $n$-vertex graph $G$ from the class can be assigned *labels* (which are binary strings) of length $f(n)$ in such a way that the decoder can infer the adjacency of any two vertices $u, v$ in $G$ from their mere labels. An $O(\log n)$-bit labeling scheme is easy to design for any class $\mathcal{C}$ of bounded degeneracy. From an ordering of $G \in \mathcal{C}$ witnessing degeneracy $d$, the label of each vertex stores its own index in the ordering and the indices of its at most $d$ neighbors that follow it in the order. Then the decoder just checks whether the index of one of $u, v$ is among the indices of the neighbors of the other vertex. Note that each label has size at most $(d+1)\lceil \log n \rceil$. For example, this was recently used to show that every subgraph-closed class with single-exponential speed of growth admits such a labeling scheme [3].

Adjacency labeling schemes of size $O(\log n)$ are at the heart of the recently-refuted Implicit Graph Conjecture (IGC) [14, 17]. The IGC speculated that the information-theoretic necessary condition for a hereditary graph class to have an $O(\log n)$-bit labeling scheme is also sufficient. This necessary condition comes from the observation that a string of length $O(n \log n)$ obtained by concatenating all vertex labels is an encoding of the graph. Therefore a class of graphs that admits an adjacency labeling scheme of size $O(\log n)$ contains at most $2^{O(n \log n)}$ (un)labeled $n$-vertex graphs. Graph classes with such a bound on the number of (un)labeled $n$-vertex graphs are called *factorial*. In this terminology, the IGC can be stated as follows: any hereditary factorial graph class admits an $O(\log n)$-bit adjacency labeling scheme.

The IGC has been refuted by a wide margin; in a breakthrough work, Hatami and Hatami [13] showed that there are factorial hereditary graph classes for which any adjacency labeling scheme requires labels of length $\Omega(\sqrt{n})$. However, the refutation is based on a counting argument and does not pinpoint an explicit counterexample. There are a number

[1] not to be confused with the *chromatic* number
[2] Throughout the paper, log denotes the logarithm function in base 2.

of explicit factorial graph classes that could refute the IGC, but the conjecture is still open for these classes. Let us call EIGC (for Explicit Implicit Graph Conjecture) this very challenge. For instance, whether the IGC holds within intersection graphs of segments, unit disks, or disks in the plane, and more generally semi-algebraic graph classes, is unsettled. Despite the workable definitions of these classes, the geometric representations alone cannot lead to $O(\log n)$-bit labeling schemes [16]. If such labeling schemes exist, they are likely to utilize some non-trivial structural properties of these graphs.

The graph parameter *symmetric difference* was introduced to design a candidate to explicitly refute the IGC [2]. A graph $G$ has symmetric difference at most $d$ if in every induced subgraph of $G$ there is a pair of vertices $u, v$ such that there are at most $d$ vertices different from $u$ and $v$ that are adjacent to exactly one of $u, v$. In other words, $u$ and $v$ are *$d$-twins*, i.e., they become twins after removing at most $d$ vertices from the graph. One can construe symmetric difference as a dense analogue of the first definition of degeneracy given above. Symmetric difference is a hereditary graph parameter: it can only decrease when taking induced subgraphs. Like classes of bounded degeneracy, classes of bounded symmetric difference are factorial [2]. Symmetric difference generalizes degeneracy in the sense that any class of graphs of bounded degeneracy has bounded symmetric difference. Indeed, if a graph has degeneracy at most $d$, then it has symmetric difference as most $2d$: for any graph with an ordering witnessing degeneracy $d$, the first two vertices in the order are $2d$-twins. Notice that, on the other hand, complete graphs have unbounded degeneracy, but their symmetric difference is 0. Classes of bounded symmetric difference contains classes of bounded twin-width, and this containment is strict as twin-width is unbounded on degenerate graphs [5]. The existence of an $O(\log n)$-bit adjacency labeling scheme for graphs of bounded symmetric difference remains open.

**Our contribution.** We introduce another dense analogue of degeneracy based on the second given definition. The *sd-degeneracy* (for *symmetric-difference degeneracy*) of a graph $G$ is the least integer $d$ for which there is an ordering of the vertices of $G$ such that every vertex $v$ but the last one admits a $d$-twin in the subgraph of $G$ induced by $v$ and all the vertices following it in the order. It follows from the definitions that graphs with sd-degeneracy at most $d$ form a superset of graphs with symmetric difference at most $d$. Contrary to what happens in the sparse setting with degeneracy, this superset is strict. In fact, there are classes with sd-degeneracy 1 and unbounded symmetric difference.

▶ **Proposition 1** ($\star$). *For any $n$-vertex graph $G$, there exists a graph of sd-degeneracy 1 with less than $n^2$ vertices containing $G$ as an induced subgraph.*

By an aforementioned counting argument, the class of all graphs requires labeling schemes of size $\Theta(n)$. Therefore, by Proposition 1, the (non-hereditary) class of graphs with sd-degeneracy at most 1 requires adjacency labels of size $\Omega(\sqrt{n})$. Surprisingly, we match this lower bound with a labeling scheme, tight up to a polylogarithmic factor, for any class of bounded sd-degeneracy.

▶ **Theorem 2.** *The class of all graphs with sd-degeneracy at most $d$ admits an $O(\sqrt{dn}\log^3 n)$-bit adjacency labeling scheme.*

The tool behind the proof of Theorem 2 is the second motivation of the paper. We wish to unify and extend twin-decompositions of low width (also called tree models) [4, 8] developed in the context of twin-width, and spanning paths (or Welzl orders) of low crossing number (or low alternation number) [22], which are useful orders in answering geometric range queries; also see [10] which utilizes these orders as part of efficient first-order model checking algorithms. We thus introduce *signed tree models*. A signed tree model of a graph $G$ is a tree whose leaves are in one-to-one correspondence with the vertices of $G$, together with extra transversal edges and anti-edges, which fully determine (see the exact rules in Section 3) the edges of $G$. The novelty compared to the existing tree models is the presence of transversal anti-edges. We show that graphs with signed tree models of degeneracy at most $d$ admit a labeling scheme as in Theorem 2. The latter theorem is then obtained by building such a signed tree model for any graph of sd-degeneracy $d$.

When given the vertex ordering witnessing sd-degeneracy $d$, the labeling scheme can be effectively computed. However, we show that computing the sd-degeneracy of a graph (hence, in particular a witnessing order) is NP-complete, even when the sd-degeneracy is guaranteed to be below a fixed constant. In the language of parameterized complexity, sd-degeneracy is para-NP-complete.

▶ **Theorem 3.** *Deciding if a graph has sd-degeneracy at most 1 is NP-complete.*

We show that, surprisingly, the other dense analogue of degeneracy, symmetric difference, is co-NP-complete. Again, the associate parameterized problem is para-co-NP-complete.

▶ **Theorem 4.** *Deciding if a graph has symmetric difference at most 8 is co-NP-complete.*

This is curious because sd-degeneracy and symmetric difference similarly extend to the dense world two equivalent definitions of degeneracy. Nevertheless, one can explain the apparent tension between Theorems 3 and 4: a vertex ordering witnesses an upper bound in the sd-degeneracy, whereas an induced subgraph witnesses a lower bound in the symmetric difference. We leave as an open question whether classes of bounded symmetric difference have labeling schemes of (poly)logarithmic size. This is excluded for bounded sd-degeneracy, for which we now know essentially optimal labeling schemes. While not an absolute barrier, the likely absence of polynomial certificates tightly upper bounding the symmetric difference complicates matters in settling this open question.

**Organization.** Section 2 gives definitions and notation. In Section 3 we introduce signed tree models, and prove that graphs of bounded sd-degeneracy admit signed tree models of bounded width. In Section 4 we show how to balance these signed tree models, and complete the proof of Theorem 2. In Section 5 we prove Theorem 4, and we prove Theorem 3 in the long version [23]. The proofs marked with a ⋆ have been moved to the appendix.

## 2 Preliminaries

We denote by $[i, j]$ the set of integers that are at least $i$ and at most $j$, and $[i]$ is a shorthand for $[1, i]$. We follow standard asymptotic notation throughout, and additionally by $f(n) = \tilde{O}(g(n))$ we mean that there exists constants $c, n_0 > 0$ such that for any $n \geqslant n_0$ we have $f(n) \leqslant g(n) \log^c n$.

We denote by $V(G)$ and $E(G)$ the vertex set and edge set of a graph $G$, respectively. Given a vertex $u$ of a graph $G$, we denote by $N_G(u)$ the set of neighbors of $u$ in $G$ (*open neighborhood*) and by $N_G[u]$ the set $N_G(u) \cup \{u\}$ (*closed neighborhood*). When $H, G$ are two graphs, we may denote by $H \subseteq_i G$ (resp. $H \subseteq G$) the fact that $H$ is an induced subgraph (resp. subgraph) of $G$, i.e., can be obtained by removing vertices of $G$ (resp. by removing vertices and edges of $G$). We denote by $G[S]$ the subgraph of $G$ induced by $S$, formed by removing every vertex of $V(G) \setminus S$. We use $G - S$ as a shorthand for $G[V(G) \setminus S]$, and $G - v$, for $G - \{v\}$.

Given two sets $A$ and $B$, we denote by $A \triangle B$ their symmetric difference, that is, $(A \setminus B) \cup (B \setminus A)$. Given a graph $G$, and two distinct vertices $u, v \in V(G)$, we set

$$\mathrm{sd}_G(u,v) := |(N_G(u) \setminus \{v\}) \triangle (N_G(v) \setminus \{u\})|.$$

The *symmetric difference* of $G$, $\mathrm{sd}(G)$, is defined as $\max_{H \subseteq_i G} \min_{u \neq v \in V(H)} \mathrm{sd}_H(u,v)$. Symmetric difference was implicitly introduced in [2] and later explicitly defined in [1]. For example, if $G$ is a planar graph, one can find in any induced subgraph $H$ of $G$ two vertices of degree less than 6. Hence planar graphs have symmetric difference bounded by 12. We call *sd-degeneracy* of $G$, denoted by $\mathrm{sdd}(G)$, the smallest non-negative integer $d$ such that $|V(G)| = 1$ or there is a pair $u \neq v \in V(G)$ satisfying $\mathrm{sd}_G(u,v) \leqslant d$ and $G - v$ has sd-degeneracy at most $d$. We say that an ordering $v_1, v_2, \ldots, v_n$ of the vertices of $G$ witnesses that the *sd-degeneracy* of $G$ is at most $d$ if for every $i \in [n-1]$, there is a $j > i$ such that $\mathrm{sd}_{G - \{v_k \ : \ k \in [i-1]\}}(v_i, v_j) \leqslant d$. It thus holds that for any graph $G$, $\mathrm{sdd}(G) \leqslant \mathrm{sd}(G)$, since for every $i \in [n]$, $G - \{v_k \ : \ k \in [i-1]\}$ is an induced subgraph of $G$. But, as shown by Proposition 1, there are some graphs with sd-degeneracy 1 and unbounded symmetric difference.

Two vertices $u, v$ are said to be *$d$-twins* in a graph $G$ if they are distinct and $|(N_G(u) \setminus N_G[v]) \cup (N_G(u) \setminus N_G[v])| \leqslant d$. The *$a \times b$ rook graph* has vertex set $\{(i,j) \ : \ i \in [a], j \in [b]\}$ and edge set $\{(i,j)(k,\ell) \ : \ (i,j) \neq (k,\ell), \ i = k \text{ or } j = \ell\}$. Equivalently it is the line graph of the bipartite complete graph $K_{a,b}$. For every $a, b \geqslant 3$, the symmetric difference of the $a \times b$ *rook graph* is $2(\min(a,b) - 1)$.

We will extensively use *tree orders*, i.e., partial orders defined by ancestor–descendant relationships in a rooted tree. We denote by $\prec_T$ the corresponding relation in rooted tree $T$. That is, $u \prec_T u'$ means that $u$ is a *strict ancestor* of $u'$ in $T$, and $u \preceq_T u'$ means that $u$ is an *ancestor* of $u'$, i.e., $u = u'$ or $u \prec_T u'$. We extend this partial order to elements of $\binom{V(T)}{2}$. An unordered pair $uv$ is an *ancestor* of $u'v'$ in $T$, denoted by $uv \preceq_T u'v'$, whenever either $u \preceq_T u'$ and $v \preceq_T v'$, or $v \preceq_T u'$ and $u \preceq_T v'$ holds. We write $uv \prec_T u'v'$ when $uv \preceq_T u'v'$ and $\{u,v\} \neq \{u',v'\}$. A rooted binary tree is *full* if all its *internal nodes*, i.e., non-leaf nodes, have exactly two children. A rooted binary tree is *complete* if all its levels are completely filled, except possibly the last one, wherein leaves are left-aligned. The *depth* of a rooted tree is the maximum number of nodes in a root-to-leaf path. We denote by $L(T)$ the set of leafs of $T$.

## 3 Signed tree models

An unordered pair of vertices in $T$ that is not in an ancestor–descendant relationship is called a *transversal pair of $T$*. Two transversal pairs $uv, u'v'$ of $T$ *cross* if $u, v$ have the same common ancestor as $u', v'$ do, and neither $uv$ is an ancestor of $u'v'$, nor $u'v'$ is an ancestor of $uv$. A *signed tree model* $\mathcal{T}$ is a triple $(T, A(T), B(T))$, where $T$ is a full binary tree, $A(T)$ (for Android green, or Anti) is a set of transversal pairs of $T$, called *transversal anti-edges*, and $B(T)$ (for Blue, or Biclique) is a set of transversal pairs of $T$, called *transversal edges*, such that $A(T) \cap B(T) \neq \emptyset$ and no $uv, u'v' \in A(T) \cup B(T)$ cross. We may refer to the transversal anti-edges as *green edges*, and to the transversal edges as *blue edges*.

The *width* of the signed tree model $(T, A(T), B(T))$ is the degeneracy of the graph $(V(T), A(T) \cup B(T))$. Note that if $(V(T), A(T) \cup B(T))$ is $d$-degenerate, then $(V(T), A(T) \cup B(T) \cup E(T))$ is $(d+3)$-degenerate. The signed tree model is $d$-sparse if $|A(T) \cup B(T)| \leqslant d|V(T)|$. We observe that a signed tree model of width $d$ is $d$-sparse, but an $O(1)$-sparse signed tree model can have width $\Omega(\sqrt{|V(T)|})$ (think of the disjoint union of a clique on $\sqrt{n}$ vertices with a set $n - \sqrt{n}$ independent vertices).

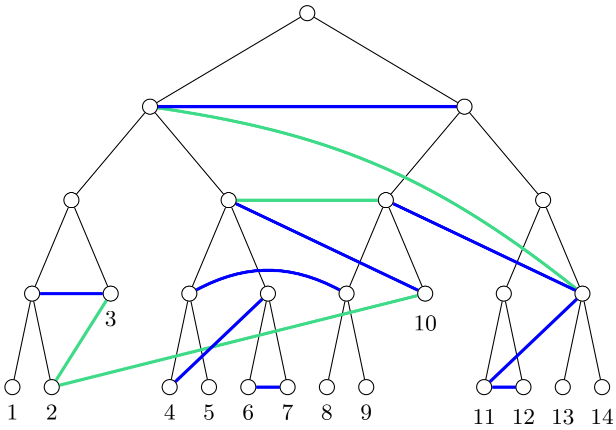


**Figure 1** A signed tree model of a 14-vertex graph.

The signed tree model $\mathcal{T} := (T, A(T), B(T))$ defines a graph $G := G_{\mathcal{T}}$ with vertex set $L(T)$. Two leaves $u, v \in L(T)$ are adjacent in $G$ if there is $u'v' \in B(T)$ such that $u' \preceq_T u$ and $v' \preceq_T v$, and there is no $u''v'' \in A(T)$ with $u'v' \prec_T u''v'' \preceq_T uv$. For example, in the representation of Figure 1, vertices 4 and 8 are adjacent in $G$ because of the blue edge between their parents (below the green edge between their grandparents), but vertices 7 and 8 are non-adjacent because of the green edge between their grandparents (below the blue edge between their great-grand-parents). We may say that a graph $G$ admits (or has) a signed tree model of width $d$ if there is a signed tree model of this width that defines $G$. Every graph $G$ admits a signed tree model as one can simply set $A(T) := \emptyset$, $B(T) := E(G)$ on an arbitrary full binary tree $T$ with $L(T) = V(G)$. However this representation may have large width, while a more subtle one (linking nodes higher up in the tree) may have a lower width.

A signed tree model is said to be *clean* if every pair of siblings are linked by a green or blue edge. It is easy to turn a signed tree model into a clean one representing the same graph: simply add green edges between every pair of siblings that were previously not linked (by a blue or green edge). This operation may only increase the width of the signed tree model by 1. The advantage of working with a clean signed tree model is that for every pair of leaves $u, v$ with least common ancestor $w$, there is at least one transversal edge or anti-edge connecting the paths (in $T$) between $w$ and $u$ and between $w$ and $v$. Clean tree models will be useful in Section 4 when we balance the trees associated with the tree models.

Given a clean signed tree model $(T, A(T), B(T))$ and $u, v \in L(T)$, we denote by $e_T(u, v)$ the unique green or blue edge $u'v'$ such that $u'v' \preceq_T uv$ and no green or blue edge $u''v''$ satisfies $u'v' \prec_T u''v'' \preceq_T uv$. The edge $e_T(u, v)$ exists because the signed tree model is clean, and is unique because no green or blue edges may cross (or be equal). Then, $u, v$ are adjacent in $G$ if and only if $e_T(u, v) \in B(T)$, i.e., $e_T(u, v)$ is a blue edge. We first show that graphs of bounded sd-degeneracy (and in particular, of bounded symmetric difference) admit clean signed tree models of bounded width.

▶ **Lemma 5.** *Any graph of sd-degeneracy $d$ admits a clean signed tree model of width $d + 1$.*

**Proof.** Let $v_1, \ldots, v_n$ be a vertex ordering that witnesses sd-degeneracy $d$ for an $n$-vertex graph $G$. For $i \in [n]$, let $G_i := G - \{v_j \ : \ 1 \leqslant j \leqslant i-1\}$. In particular, $G_1 = G$. Let $u_i$ be a $d$-twin of $v_i$ in $G_i$. Initially we consider a forest of $n$ distinct 1-vertex rooted trees, each root labeled by a distinct vertex of $G$. We will build $T$ (and in parallel, the transversal anti-edges and edges) by iteratively giving a common parent to two roots of this forest of $n$ singletons. Note that different nodes of $T$ may have the same label, as the labels will range in $V(G)$ whereas $T$ has $2n-1$ nodes.

For $i$ ranging from 1 to $n-1$:

- add a blue (resp. green) edge between $v_i$ and $u_i$ if $u_iv_i \in E(G)$ (resp. $u_iv_i \notin E(G)$),
- add a blue edge between $v_i$ and the roots labeled by $w$ for $w \in N_{G_i}(v_i) \setminus N_{G_i}[u_i]$,
- add a green edge between $v_i$ and the roots labeled by $w$ for $w \in N_{G_i}(u_i) \setminus N_{G_i}[v_i]$, and
- create a common parent, labeled by $u_i$, for the roots labeled $u_i$ (left child) and $v_i$ (right child).

This defines a full binary tree $T$ such that $L(T) = V(G)$. In $(V(T), A(T) \cup B(T))$, the leaves labeled by $v_1$ and $u_1$ have degree at most $d+1$ and 1, respectively. Hence an immediate induction on $(T, A(T), B(T))$ (after removing these two leaves, and following the order $v_2, \ldots, v_n$) shows that $(V(T), A(T) \cup B(T))$ is $(d+1)$-degenerate. As we only add transversal anti-edges and edges between pairs of roots, no pair in $A(T) \cup B(T)$ can cross. Indeed if $x, y$ are two nodes of $T$ that are both roots in some $G_i$, then it cannot happen that $x', y'$ are also both roots of some $G_{i'}$ with $x \prec_T x'$ and $y' \prec_T y$. The first item further ensures that the signed tree model $(T, A(T), B(T))$ of width $d+1$ is clean.

Let us finally check that for every $u, v \in L(T)$, $e_T(u, v)$ is a blue edge if and only if $uv \in E(G)$. This is a consequence of the following property.

▷ Claim 6. Let $x, y$ be two nodes of $T$ labeled by $u, v$ respectively. Let $x'$ be a child of $x$, labeled by $u'$, such that $x'y$ is neither a blue nor a green edge. Further assume that $y$ was a root when the parent of $x'$ (i.e., $x$) was created. Then, $uv \in E(G)$ if and only if $u'v \in E(G)$.

Proof. If $x'$ is the left child of $x$, the conclusion holds since $u = u'$. We can thus assume that $x'$ is the right child of $x$, and not the sibling of $y$ since it would contradict that $x'y$ is neither a blue nor a green edge. Node $x'$ was not linked to $y$ by a blue or a green edge, so $v$ cannot be a neighbor of exactly one of $u, u'$. ◁

Consider the moment $e_T(u, v)$ was added to the signed tree model, say between the then-roots $x$ and $y$, labeled by $u'$ and $v'$, respectively. By the way blue and green edges are introduced, $xy$ is a blue edge if $u'v' \in E(G)$, and $xy$ is green if $u'v' \notin E(G)$. Thus we conclude by iteratively applying Claim 6. ◀

## 4 Balancing Signed Tree Models

For any signed tree model of width $d$ of an $n$-vertex graph, we get an adjacency labeling scheme with labels of size $O(dh \log n)$, where $h$ is the depth of $T$. Indeed, one can label a leaf $v$ of $T$ (i.e., vertex of $G$) by the identifiers (each of $\log(2n)$ bits) of all the nodes of the path from $v$ to the root of $T$, followed by the identifiers of the outneighbors of these at most $h$ nodes in a fixed orientation of $(V(T), A(T) \cup B(T))$ with maximum outdegree at most $d+1$, allocating an extra bit for the color of each corresponding edge. One can then decode the adjacency of any pair $u, v \in V(G)$ by looking at the color of $e_T(u, v)$. The latter is easy to single out, based on the labels of $u$ and $v$.

▶ **Proposition 7.** *Let $G$ be an $n$-vertex graph with a signed tree model of width $d$ and depth $h$. Then, $G$ admits an $O(dh \log n)$-bit adjacency labeling scheme.*

Unfortunately, the depth of the tree $T$ of a signed tree model of low width obtained for an $n$-vertex graph of low sd-degeneracy could be as large as $n$. This makes a direct use of Proposition 7 inadequate. Instead, we first decrease the depth of the signed tree model, while controlling its sparsity. We rely on the following simple lemma.

▶ **Lemma 8.** *Let $T$ be a full, complete tree, whose leaves read $1, \ldots, n \geqslant 2$ from left to right. Any interval $[i, j]$ with $i, j \in [n]$ is the disjoint union of the leaves of at most $2 \log n$ rooted subtrees of $T$.*

**Proof.** Let $X \subseteq V(T)$ be such that the leaves of the subtrees rooted at a node of $X$ partition $[i, j]$, and $X$ is of minimum cardinality among node subsets with this property. Let $k$ be the first level of $T$ intersected by $X$ (with the root being at level 1). At most two nodes $x, y$ of $X$ are at level $k$ (and exactly one node when $k = 2$), with $x = y$ or $x$ to the left of $y$. Observe that if $x \neq y$, then $x, y$ have to be consecutive along the left-to-right ordering of level $k$, but cannot be siblings (otherwise they can be substituted by their parent). At level $k + 1$, at most two nodes can be part of $X$: the node just to the left of the leftmost child of $x$, and the node just to the right of the rightmost child of $y$. This property propagates to the last level. Thus $|X| \leqslant \max(2(\lceil \log n \rceil - 1), \lceil \log n \rceil) \leqslant 2 \log n$. Note indeed that there are $\lceil \log n \rceil + 1$ levels. ◀

From the previous proof it can also be seen that there is a unique minimum-cardinality set $X$ representing $[i, j]$. In what follows, let us denote it by $X_{i,j}$. We also denote by $I_T(x)$ the set of leaves of the subtree of $T$ rooted at $x \in V(T)$.

▶ **Observation 9.** *For every rooted tree $T$, and every $x, y \in V(T)$, if $I_T(x)$ and $I_T(y)$ intersect, then one is included in the other.*

We are now ready to prove the main lemma of this section.

▶ **Lemma 10.** *Let $(T, A(T), B(T))$ be a clean $d$-sparse signed tree model of an $n$-vertex graph $G$. Then, $G$ admits a $4d \log^2 n$-sparse signed tree model $(T', A(T'), B(T'))$ of depth $\lceil \log n \rceil + 1$.*

**Proof.** Consider the left-to-right order on $L(T)$. To ease the notation, say that the leaves are labeled $1, 2, \ldots, n$ in this order. We choose for $T'$ the full, complete binary tree whose leaves are also labeled by $1, 2, \ldots, n$ when read from left to right. For every transversal anti-edge (resp. edge) $xy \in A(T)$ (resp. $xy \in B(T)$), note that $I_T(x)$ and $I_T(y)$ are discrete intervals. Let $[i, j] := I_T(x)$ and $[i', j'] := I_T(y)$. We add to $A(T')$ (resp. $B(T')$) all the unordered pairs $x'y'$ with $x' \in X_{i,j}$ and $y' \in X_{i',j'}$. It may happen that some $x'y'$ is added both to $A(T')$ and $B(T')$. In which case, $x'y'$ originates from both $x_0y_0 \in A(T)$ and $x_1y_1 \in B(T)$ such that $x_0y_0 \prec_T x_1y_1$ or $x_1y_1 \prec_T x_0y_0$. In the former case, we remove $x'y'$ from $B(T')$ (and only keep it in $A(T')$), and in the latter, we remove $x'y'$ from $A(T')$ (and only keep it in $B(T')$). This finishes the construction of $\mathcal{T}' := (T', A(T'), B(T'))$.

Let us first argue that no pairs of green or blue edges cross in $T'$. Assume for the sake of contradiction that $a'b', c'd' \in A(T') \cup B(T')$ satisfy $a' \prec_{T'} c'$ and $d' \prec_{T'} b'$. Let $ab, cd \in A(T) \cup B(T)$ be the green or blue edges that created $a'b', c'd'$, respectively. As $I_T(a) \supseteq I_{T'}(a')$, $I_T(c) \supseteq I_{T'}(c')$, and $a' \prec_{T'} c'$, $I_T(a)$ and $I_T(c)$ intersect. Thus by Observation 9, $I_T(a) \subseteq I_T(c)$ or $I_T(c) \subseteq I_T(a)$. By minimality of the sets $X_{i,j}$, $I_T(c)$ cannot include $I_{T'}(a')$. Thus $I_T(c) \subset I_T(a)$, so $a \prec_T c$. Analogously $d \prec_T b$, which implies that $ab$ and $cd$ cross in $T$. Therefore $\mathcal{T}'$ is a signed tree model.

By design, the depth of $T'$ is $\lceil \log n \rceil + 1$. As $\mathcal{T} := (T, A(T), B(T))$ is $d$-sparse, it has at most $(2n - 1)d$ transversal (anti-)edges. Each blue or green edge of $\mathcal{T}$ gives rise to at most $(2 \log n)^2$ blue or green edges of $\mathcal{T}'$, by Lemma 8. Hence $\mathcal{T}'$ is $4d \log^2 n$-sparse.

Let us finally check that $\mathcal{T}'$ still represents $G$. Fix $u, v \in V(G)$ and $xy := e_T(u, v)$. Let $x'y'$ be the green or blue edge of $\mathcal{T}'$ originating from $xy$ such that $u \in I_{T'}(x'), v \in I_{T'}(y')$. We claim that $x'y'$ cannot have been removed (see the technicality at the end of the construction of $\mathcal{T}'$), nor can $x''y'' \in A(T') \cup B(T')$ hold with $x'y' \prec_T x''y''$. Indeed, by the arguments of the second paragraph, the green or blue edge $e$ of $\mathcal{T}$ giving rise to $x''y''$ would be such that $xy \prec_T e \preceq_T uv$, contradicting the definition of $e_T(u, v)$. ◀

We finally need this folklore observation.

▶ **Observation 11.** *Every $m$-edge graph has degeneracy at most $\lceil \sqrt{2m} \rceil - 1$.*

**Proof.** It is enough to show that any $m$-edge graph $G$ has a vertex of degree at most $\lceil \sqrt{2m} \rceil - 1$. If all the vertices of $G$ have degree at least $\lceil \sqrt{2m} \rceil$, then $m \geqslant \frac{1}{2}n\lceil \sqrt{2m} \rceil$. But also $n \geqslant \lceil \sqrt{2m} \rceil + 1$ for a vertex to possibly have $\lceil \sqrt{2m} \rceil$ neighbors. Thus $m \geqslant \frac{1}{2}\sqrt{2m}(\sqrt{2m}+1) > m$, a contradiction. ◀

Combining Lemmas 5 and 10, Observation 11, , and Proposition 7 yields Theorem 2.

**Proof of Theorem 2.** Let $G$ be an $n$-vertex graph of sd-degeneracy $d$. By Lemma 5, $G$ admits a clean signed tree model of width at most $d + 1$, hence $(d + 1)$-sparse. Thus by Lemma 10, $G$ has a $4(d + 1)\log^2 n$-sparse signed tree model $\mathcal{T}$ of depth $\lceil \log n \rceil + 1$. By Observation 11, $\mathcal{T}$ has width at most

$$\sqrt{16(d+1)n\log^2 n} = 4\sqrt{(d+1)n}\log n.$$

Therefore, by Proposition 7, $G$ has a $O(\sqrt{dn}\log^3 n)$-bit labeling scheme. ◀

## 5 Symmetric Difference is para-co-NP-complete

For any fixed even integer $d \geqslant 8$, we show that the following problem is NP-complete: Does the input graph $G$ have an induced subgraph with at least two vertices and no pair of $d$-twins? We call such an induced subgraph a $(d + 1)$-*diverse graph*. The membership of this problem to NP is straightforward, as a $(d+1)$-diverse induced subgraph $H$ of $G$ is a polynomial-sized witness. One can indeed check in polynomial-time that $H$ has at least two vertices, and that for every pair $u, v$ of vertices of $H$, at least $d+1$ other vertices of $H$ are neighbors of exactly one of $u, v$.

The *$d$-twin graph* $T_d(G)$ of a graph $G$ is a graph with vertex set $V(G)$ and edges between every pair of $d$-twins.

▶ **Observation 12.** *The vertices of a $(d+1)$-diverse induced subgraph of $G$ form an independent set of $T_d(G)$.*

Given any 3-SAT formula $\varphi$ with at most three occurrences of each variable, clauses of size two or three, and at least three clauses, we build a graph $G := G(\varphi)$ such that $G$ has a $(d + 1)$-diverse induced subgraph if and only if $\varphi$ is satisfiable. Such a restriction of 3-SAT is known to be NP-complete [21].

### 5.1 Bubble gadget

A *bubble gadget* $B$ (or *bubble* for short) is a $w \times w$ rook graph, with $w := \frac{d}{2} + 2$, deprived of the two rightmost vertices of its top row. We say that $B$ is *properly attached* to the rest of the graph if each vertex of the top row (of width $\frac{d}{2}$) and of the rightmost column (of height $\frac{d}{2} + 1$) has one or two neighbors outside the gadget, whereas the other vertices of $B$ have no neighbors outside $V(B)$. Let $S$ be the set of neighbors of the bubble outside of $B$.

We say that $B$ is *neatly attached to $S$* if it is properly attached to $S$, and further, vertices of the top row and rightmost column have *exactly* one outside neighbor, and at most one vertex of $S$ has neighbors in both the top row and rightmost column. The *neat* attachments that we will use, in this section and the next, satisfy $2 \leqslant |S| \leqslant 5$. Hence they can be described by a tuple of size between 2 and 5, listing the number of neighbors of vertices in $S$ among $V(B)$, starting with the top row and ending with the rightmost column. For instance, Figure 2 depicts a neat $(2, 2, 2, 7)$-attachment. A bubble properly attached to $S$ is in a delicate state. It may entirely survive in a $(d + 1)$-diverse induced subgraph of $G$.

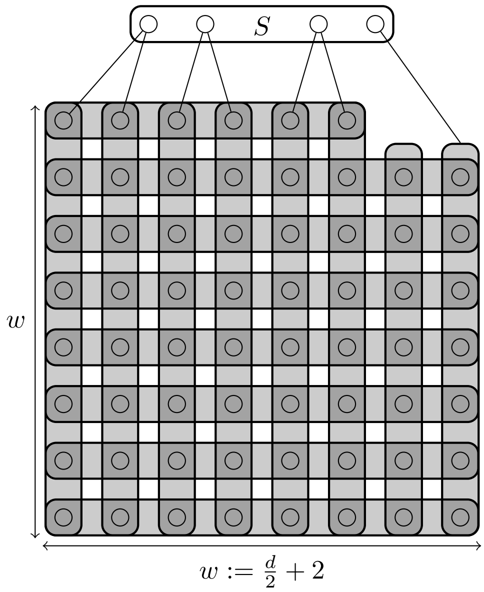


**Figure 2** A neatly $(2,2,2,7)$-attached bubble gadget, with $d = 12$.

▶ **Observation 13.** *Let $B$ be a bubble gadget properly attached to $S$ in $G$. No pair of vertices of $B$ are $d$-twins in $G[V(B) \cup S]$.*

**Proof.** In $B$ the only pairs with symmetric difference at most $d$, in fact exactly $d$, consist of a vertex in the top row and another vertex in its column, or two vertices of the same row in the two rightmost columns. In both cases, these pairs have symmetric difference at least $d+1$ in $G[V(B) \cup S]$ since vertices of the top row or rightmost column have at least one neighbor in $S$, while all other vertices of $B$ have no neighbor in $S$. ◀

However, deletions that cause one vertex of the top row or two vertices of the rightmost column to no longer have outside neighbors cause the bubble to completely collapse.

▶ **Lemma 14** (⋆)**.** *Let $B$ be a bubble gadget properly attached to $S$ in $G$. Let $H$ be any $(d+1)$-diverse induced subgraph of $G$, such that at least one vertex of the top row or at least two vertices of the rightmost column has no neighbor in $V(H) \setminus V(B)$. Then, $H$ contains at most one vertex of $B$.*

In the current section, for the hardness of symmetric difference, all the bubble gadgets will be neatly attached. Furthermore, every vertex a bubble is attached to will have at least one neighbor on the top row, or at least two neighbors in the rightmost column. Thus the deletion of *any* vertex a bubble $B$ is attached to will result, by Lemma 14, in deleting all the vertices of $B$ but at most one.

## 5.2 Variable and clause gadgets

The *variable gadget* of variable $x$ used in $\varphi$ is simply two vertices $x, \neg x$ adjacent to a set $N_x$ of $t := \frac{d}{2} + 1$ shared neighbors. Since each literal appears positively and negatively in $\varphi$ (otherwise the valuation of the literal is clear), vertices $x$ and $\neg x$ have one or two other neighbors in $G$ corresponding to the clause they belong to, as we will soon see. The vertices of $N_x$ will have other neighbors split into at most four bubble gadgets. This too will be

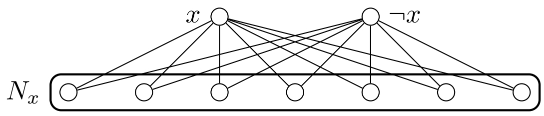


**Figure 3** The variable gadget of $x$ with $d = 12$.

described shortly. The *clause gadget* of clause $c$ consists of a pair of adjacent vertices $v_c, d_c$. We make $v_c$ (but not $d_c$) adjacent to the two or three vertices corresponding to the literals of $c$.

## 5.3 Construction of $G(\varphi)$

Unsurprisingly, we add one variable gadget per variable, and one clause gadget per clause of $\varphi$. Let $x_1, \dots, x_n$ be a numbering of the variables, and $c_1, \dots, c_m$, of the clauses. We neatly attach a bubble gadget to $S_j$ made of the five vertices $z_j, v_{c_j}, d_{c_j}, v_{c_{j+1}}, d_{c_{j+1}}$ for every $j \in [m-1]$, with (in this order) a $(1, \lfloor d/4 \rfloor, \lfloor d/4 \rfloor, \lceil d/4 \rceil, \lceil d/4 \rceil)$-attachment, where $z_j$ is a vertex of some $N_x$. The choice of $z_j$ is irrelevant, but we take all the vertices $z_j$ pairwise distinct. This is possible since there are at most $3n/2$ clauses, and more than $dn/2$ vertices contained in the union of the sets $N_x$. For every $j \in [m-1]$, we make $\{v_{c_j}, d_{c_j}\}$ fully adjacent to $\{v_{c_{j+1}}, d_{c_{j+1}}\}$. The construction of $G$ is almost complete; see Figure 4 for an illustration.

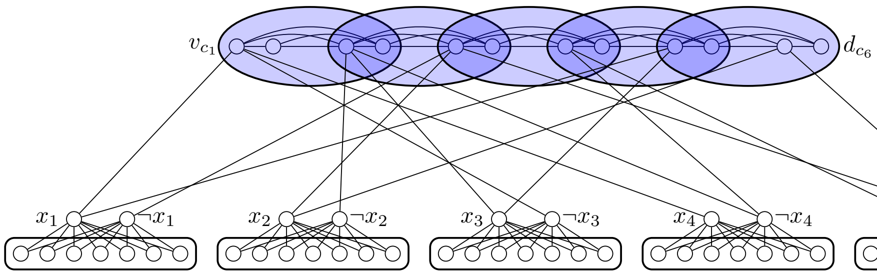


**Figure 4** The essential part of $G$ built so far, for a 3-CNF formula $\varphi$ whose first two clauses are $x_1 \vee \neg x_3 \vee x_4$ and $\neg x_2 \vee x_3 \vee \neg x_4$. The blue ellipses represent the bubbles attached to the four enclosed vertices (recall that the bubble is attached to a fifth vertex among the sets $N_x$).

At this point, all the vertices $v_{c_j}, d_{c_j}$ such that $j \in [2, m-1]$ have exactly $\lfloor d/4 \rfloor + \lceil d/4 \rceil = d/2$ neighbors in (two) bubble gadgets. Let $y_1, \dots, y_{nt}$ be the ordering of the vertices in $\bigcup_x N_x$ from left to right in how they appear in Figure 4. We neatly attach a bubble gadget to $(v_{c_1}, d_{c_1}, y_1)$ by a $(\lceil d/4 \rceil, \lceil d/4 \rceil, d+1-2\lceil d/4 \rceil)$-attachment. Similarly, we neatly attach a bubble gadget to $(v_{c_m}, d_{c_m}, y_{nt})$ by a $(\lceil d/4 \rceil, \lceil d/4 \rceil, d+1-2\lceil d/4 \rceil)$-attachment. Finally for every $i \in [nt-2]$, we neatly attach a bubble gadget to $S'_i$ made of the three vertices $y_i, y_{i+1}, y_{i+2}$ with a $(1, d/2, d/2)$-attachment. This finishes the construction.

We make some observations. As all the bubble gadgets are neatly attached, no two vertices outside a bubble gadget $B$ can share a neighbor in $B$.

▶ **Observation 15.** *For every $j \in [m]$, $v_{c_j}, d_{c_j}$ each have exactly $d/2$ neighbors in bubble gadgets (all of which are non-adjacent to any other vertex outside their respective bubble).*

Vertices in $\bigcup_x N_x$ have more neighbors in bubbles.

▶ **Observation 16.** *Every $v \in \bigcup_x N_x$ has at least $d/2+1$ neighbors in bubble gadgets (all of which are non-adjacent to any other vertex outside their respective bubble).*

## 5.4 Correctness

We can now show the following strengthening of Theorem 4.

▶ **Theorem 17.** *For every fixed even integer $d \geqslant 8$, deciding if an input graph has symmetric difference at most $d$ is co-NP-complete.*

As the graph $G := G(\varphi)$ presented in Section 5.3 can be constructed in polynomial time, we shall simply check the equivalence between the satisfiability of $\varphi$ and the existence of a $(d+1)$-diverse induced subgraph of $G(\varphi)$. We recall that, by definition, $G$ has *not* symmetric difference at most $d$ if and only if it has a $(d+1)$-diverse induced subgraph.

▶ **Lemma 18.** *If $\varphi$ is satisfiable, then $G$ admits a $(d+1)$-diverse induced subgraph.*

**Proof.** Let $\mathcal{A}$ be a satisfying assignment of $\varphi$. For each variable $x$ of $\varphi$, we delete vertex $\neg x$ if $\mathcal{A}$ sets $x$ to true, and we delete vertex $x$ otherwise (if $\mathcal{A}$ sets $x$ to false). Let us call $H$ the obtained induced subgraph of $G$ (with at least two vertices). We claim that $H$ has no pair of $d$-twins, and successively rule out such pairs

**(i)** within the same bubble,
**(ii)** between a vertex in a bubble $B$ and a vertex outside $B$ (but possibly in another bubble),
**(iii)** between two vertices both outside every bubble gadget.

**(i)** As $H$ contains all the vertices of $G$ on which bubble gadgets are attached, by Observation 13, no two distinct vertices in the same bubble are $d$-twins.

**(ii)** Let us fix a bubble gadget $B$ attached to $S$, and two vertices $u \in V(B)$ and $v \in V(H) \setminus V(B)$. First observe that $u$ has at least $d/2+1$ neighbors in $V(B)$ (hence in $H$) that are not neighbors of $v$. All the vertices $v \in V(H) \setminus (V(B) \cup S)$ have at least $d/2$ neighbors in $H$ that are not neighbors of $u$. For these vertices $v$, $\mathrm{sd}_H(u,v) > d$. We thus focus on the case when $v \in S$. We can assume that $v$ is some $v_{c_j}$ or $d_{c_j}$, as any other vertices have at least $d/2$ neighbors outside $B$. We can further assume that $u$ is in the top row or rightmost column of $B$, otherwise it has $d$ neighbors that are not neighbors of $u$ (and $u$ has at least one private neighbor). Now we observe that

$$\mathrm{sd}_H(u,v) \geqslant |N_H(u) \setminus N_H[v]| + |N_H(v) \setminus N_H[u]| \geqslant d/2+1+d/2-1-\lceil d/4 \rceil + \lfloor d/4 \rfloor + 2 \geqslant d+1,$$

where $d/2+1$ lower bounds the number of neighbors of $u$ whose neighborhood is included in $V(B)$, $d/2-1-\lceil d/4 \rceil$ lower bounds the number of neighbors of $u$ in the top row or rightmost column of $B$ that are not adjacent to $v$, $\lfloor d/4 \rfloor$ lower bounds the number of neighbors of $v$ in another bubble than $B$, and 2 accounts for the at least two neighbors $v_{c_{j-1}}, d_{c_{j-1}}$ or $v_{c_{j+1}}, d_{c_{j+1}}$ of $v$, whichever exist. (Here we need that there are at least two clauses.)

**(iii)** Let $u, v$ be two distinct vertices outside every bubble gadget. Vertex $u$ (resp. $v$) has at least $d/2$ neighbors that are not neighbors of $v$ (resp. $u$). This holds by Observations 15 and 16, and the fact that every vertex $x_i$ or $\neg x_i$ is adjacent to $N_{x_i}$, while no other vertex outside the bubble gadgets is adjacent to any vertex in $N_{x_i}$. Furthermore, as $|N_{x_i}| = d/2+1$ and vertices in $\bigcup_x N_x$ have at least $d/2+1$ neighbors in bubble gadgets, the only pairs that could be $d$-twins in $H$ are made of two vertices in clause gadgets. As there are at least three clauses in $\varphi$, two vertices $u, v$ from distinct clause gadgets have at least two additional private neighbors. Thus we can assume that $u = v_{c_j}$ and $v = d_{c_j}$ for some $j \in [m]$. As $\mathcal{A}$ is a satisfying assignment, at least one vertex $x$ or $\neg x$ adjacent to $v_{c_j}$ has survived in $H$. Hence $\mathrm{sd}_H(u,v) \geqslant d/2 + d/2 + 1 = d+1$. ◀

▶ **Lemma 19** ($\star$). *If $\varphi$ is not satisfiable, then $G$ has no $(d+1)$-diverse induced subgraph.*

## 6 Discussion and open problems

Degeneracy can be defined either by degeneracy ordering for vertices, or by the existence of vertices of small degree in all the induced subgraphs. And despite sd-degeneracy and symmetric difference arising as dense counterparts to these two equivalent definitions, they are not equivalent: classes of bounded symmetric difference are strictly contained in classes of bounded sd-degeneracy. Using signed tree models, we achieve an adjacency labeling scheme for classes of bounded sd-degeneracy that is tight up to logarithmic factors. The necessity of these additional logarithmic factors remains questionable. Moreover, an optimal adjacency labeling scheme for classes of bounded symmetric difference is yet to be found.

▶ **Question 1.** *Is there an $O(\sqrt{n})$-adjacency labeling scheme for classes of bounded sd-degeneracy? Is there an $O(\log n)$-adjacency labeling scheme for classes of bounded symmetric difference?*

On the other hand, we prove a surprising phenomenon: not only both symmetric difference and sd-degeneracy lead to classes that are hard to recognize, but they respectively lead to para-NP-complete and para-co-NP complete problems. However, the existence of a polynomial-time approximation for remains open.

▶ **Question 2.** *Is there a polynomial-time algorithm to compute an approximation of symmetric difference (and sd-degeneracy)?*

### References

1 Bogdan Alecu, Aistis Atminas, and Vadim V. Lozin. Graph functionality. *J. Comb. Theory, Ser. B*, 147:139–158, 2021. doi:10.1016/j.jctb.2020.11.002.

2 Aistis Atminas, Andrew Collins, Vadim V. Lozin, and Victor Zamaraev. Implicit representations and factorial properties of graphs. *Discret. Math.*, 338(2):164–179, 2015. doi:10.1016/j.disc.2014.09.008.

3 Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev, and Maksim Zhukovskii. Tight bounds on adjacency labels for monotone graph classes. *arXiv preprint arXiv:2310.20522*, 2023.

4 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width III: max independent set, min dominating set, and coloring. In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ICALP.2021.35.

5 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. *Combinatorial Theory, 2 (2)*, 2022.

6 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, and Stéphan Thomassé. Twin-width V: linear minors, modular counting, and matrix multiplication. In *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023*, volume 254 of *LIPIcs*, pages 15:1–15:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.STACS.2023.15.

7 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022. doi:10.1145/3486655.

8 Édouard Bonnet, Jaroslav Nesetril, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations. *arXiv preprint arXiv:2102.06880*, 2021.

9 Reinhard Diestel. *Graph Theory*. Springer Berlin Heidelberg, 2017.

10 Jan Dreier, Ioannis Eleftheriadis, Nikolas Mählmann, Rose McCarty, Michal Pilipczuk, and Szymon Torunczyk. First-order model checking on monadically stable graph classes. *CoRR*, abs/2311.18740, 2023. doi:10.48550/arXiv.2311.18740.

**11** Paul Erdős and András Hajnal. On chromatic number of graphs and set-systems. *Acta Mathematica Hungarica*, 17(1-2):61–99, 1966.

**12** Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. `doi:10.1145/3051095`.

**13** Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS 2022)*, pages 1134–1137. IEEE, 2022.

**14** Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988)*, pages 334–343, 1988.

**15** Alexandr V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.

**16** Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013.

**17** John Harold Muller. *Local structure in graph classes.* PhD thesis, Georgia Institute of Technology, 1988.

**18** Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics.* Springer, 2012. `doi:10.1007/978-3-642-27875-4`.

**19** Sergey Norin, Luke Postle, and Zi-Xia Song. Breaking the degeneracy barrier for coloring graphs with no $K_t$ minor. *Advances in Mathematics*, 422:109020, 2023.

**20** Andrew Thomason. An extremal function for contractions of graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 95, pages 261–265. Cambridge University Press, 1984.

**21** Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984. `doi:10.1016/0166-218X(84)90081-7`.

**22** Emo Welzl. Partition trees for triangle counting and other range searching problems. In Herbert Edelsbrunner, editor, *Proceedings of the Fourth Annual Symposium on Computational Geometry (SoCG 1988)*, pages 23–33. ACM, 1988. `doi:10.1145/73393.73397`.

**23** Édouard Bonnet, Julien Duron, John Sylvester, and Viktor Zamaraev. Symmetric-difference (degeneracy) and signed tree models, 2024. `arXiv:2405.09011`.


## A Proof of Proposition 1

Take any $n$-vertex graph $G$, and fix an arbitrary ordering $v_1, v_2, \ldots, v_n$ of its vertices. There is a graph $G'$ with at most $(n-1)(n-3)$ vertices that both contains $G$ as an induced subgraph, and has sd-degeneracy at most 1. The graph $G'$ can be built by adding to $G$, for every $i \in [n-1]$, up to $n-3$ vertices $v_{i,1}, \ldots, v_{i,h_i}$ gradually "interpolating" between the neighborhood of $v_i$ and that of $v_{i+1}$ (in $G$). For instance, $v_{i,1}, \ldots, v_{i,h'_i}$ remove one-by-one the neighbors of $v_i$ that are not neighbors of $v_{i+1}$, and $v_{i,h'_i+1}, \ldots, v_{i,h_i}$ add one-by-one the neighbors of $v_{i+1}$ that are not neighbors of $v_i$. (We put no edge between a pair of added vertices $v_{i,p}, v_{i',p'}$.) Then an ordering witnessing sd-degeneracy at most 1 is $v_1, v_{1,1}, v_{1,2}, \ldots, v_{1,h_1}, v_2, v_{2,1}, v_{2,2}, \ldots, v_{2,h_2}, \ldots, v_{n-1}, v_{n-1,1}, v_{n-1,2}, \ldots, v_{n-1,h_{n-1}}, v_n$, for which the 1-twin of a vertex is its successor.

## B More context on signed tree models

A wide range of structural graph invariants, called width parameters, can be expressed via so-called *tree layouts* (or at least parameters functionally equivalent to them can). A *tree layout* of an $n$-vertex graph $G$ is a full binary tree $T$ such that the leaves of $T$, that we may denote by $L(T)$, are in one-to-one correspondence with $V(G)$. Width parameters are typically defined through evaluating a particular function on bipartitions of $V(G)$ made by the two connected components of $T$ when removing one edge of $T$. The width is then the minimum over tree layouts of the maximum over all such evaluations. In the definition of signed tree models, we depart from this viewpoint, and instead augment $T$ with a sparse structure encoding the graph $G$.

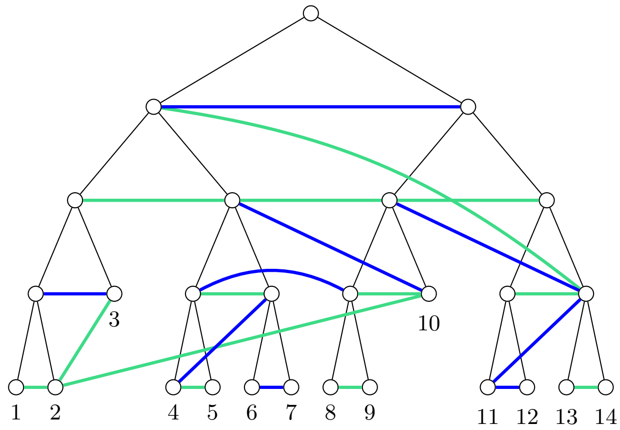


**Figure 5** The signed tree model of Figure 1 made clean.

Every graph of twin-width $d$ admits a signed tree model with $A(T) = \emptyset$ and width at most $d + 1$. *Tree models* or *twin-decompositions* are signed tree models with $A(T) = \emptyset$, and further technical requirements. We observe that similar objects to signed tree models were utilized in [6] to attain a fast matrix multiplication on matrices of low twin-width. We will not need a definition of twin-width, and refer the interested reader to [7]. In Section 1 we also mentioned Welzl orders with low alternation number [22], let us now elaborate on that.

A *Welzl order* of *alternation number $d$* for a graph $G$ is a total order $<$ on $V(G)$ such that the neighborhood of every vertex is the union of at most $d$ intervals along $<$. We claim that bipartite graphs $G = (X \uplus Y, E(G))$ with a Welzl order $<$ of alternation number $d$ admit a signed tree model of width $2d$. Note that we can assume that for every $x \in X$ and $y \in Y$, $x < y$. We build a signed tree model $(T, A(T), B(T))$ of $G$ as follows. Let us call *binary comb* a full binary tree whose internal nodes induce a path, rooted at an endpoint of this path. We make the root of $T$ adjacent to the roots of two binary combs with $|X|$ and $|Y|$ leaves, respectively. The leaves are labeled from left to right with the vertices of $G$ in the order $<$. To simplify the notations, assume that these labels describe $[n]$ in the natural order. To represent that vertex $i \in X$ has $[j, k] \subseteq Y$ in the partition of its open neighborhood into maximal intervals, we add a blue edge between leaf $i$ and the parent of $k$, and a green edge between $i$ and the parent of $j - 1$ (to stop the interval). Finally observe that $(V(T), A(T) \cup B(T))$ has maximum degree at most $2d$. (The subtree whose leaves are the vertices of $X$ need not be a binary comb.)

A similar construction would work for graphs $G$ of chromatic number $q$, and would yield a signed tree model of width $2(q-1)d$. A more permissive definition of signed tree models, allowing leaf-to-ancestor transversal edges, would give models of width $2d$ for any graph with a Welzl order of alternation number $d$. However, with this alternative definition, the consequences of Section 4 would not follow. Hence we stick to the given definition of signed tree models.

## C Proof of Lemma 14

We first deal with the case when a vertex $v$ of the top row (in the entire $B$) has no neighbor in $V(H) \setminus V(B)$. By symmetry, assume that $v$ is the topmost vertex of the first column. Vertex $v$ is thus $d$-twin with all the other vertices of the first column. Hence by Observation 12, either $v$ is not in $H$, or none of the $d/2+1$ vertices below $v$ are in $H$.

If the latter holds, then any two vertices in the same column, outside the top row and rightmost column, are now $d$-twins. By Observation 12 within these vertices, $H$ can only contain at most one vertex per column. In turn, the kept vertices are $d$-twins, so at most one can be kept overall. We conclude since the vertices of $N_H(S) \cap V(B)$ have at most two neighbors in $S$.

We now suppose that $v$ is not in $H$. Then, in each row but the topmost, the vertices in the first and penultimate columns are $d$-twins. Thus, within each pair, at most one vertex can be in $H$. This implies that any two vertices in the same column, outside the top row and rightmost column, are now $d$-twins. Thus we conclude as in the previous paragraph.

We now deal with the case when two vertices $x, y$ of the right most column have no neighbor in $V(H) \setminus V(B)$. By symmetry, we can assume that $x$ is in the second row, and $y$ is in the third row. Then $x$ (resp. $y$) is $d$-twin with the vertex just to its left. After one vertex is removed in each pair, in each column but the last two, the vertices in the second and third rows have become $d$-twins. Therefore, $H$ can only contain at most one vertex from all these pairs. We reach again the state that any two vertices in the same column, outside the top row and rightmost column, are $d$-twins, and conclude as previously.

## D Proof of Lemma 19

For each variable $x$, the vertices $x, \neg x$ are 3-twins, thus at least one of them has to be removed in a $(d+1)$-diverse induced subgraph. The kept literals (if any) define a (partial) truth assignment. By assumption, this assignment does not satisfy at least one clause $c_j$. This implies that $v_{c_j}, d_{c_j}$ are $d$-twins in the corresponding induced subgraph. Indeed, they each have exactly $d/2$ private neighbors in bubble gadgets, and no other private neighbor.

By Lemma 14, the bubble attached to $S_j$ is reduced to at most one vertex, say $w_j$ (if any). In turn, this makes the pairs $v_{c_{j-1}}, d_{c_{j-1}}$ and $v_{c_{j+1}}, d_{c_{j+1}}$ $d$-twins (when they exist). Indeed their symmetric difference is at most $3 + \lceil d/4 \rceil + 1 \leqslant d$, where 3 accounts for the three literals of the clause, and 1 for vertex $w_j$. This iteratively collapses every bubble attached to some $S_{j'}$ to a single vertex, as well as the two bubble gadgets attached to $\{v_{c_1}, d_{c_1}, y_1\}$ and $\{v_{c_m}, d_{c_m}, y_{nt}\}$, in say, $w_0$ and $w_m$. Now all the vertices $w_j$ (for $j \in [0, m]$) are 6-twins, so at most one can be kept. We recall that at most one vertex per clause gadget could be kept. For $j$ going from 1 to $m-1$, the vertex kept (if any) from the clause gadget of $c_j$ is an 8-twin of the vertex kept in the next surviving clause gadget. This implies that from all the clause gadgets and all the bubble gadgets attached to them, one can only keep at most one vertex overall, say $z$. This vertex has degree at most 3 in the resulting induced subgraph.

Vertices $y_1, y_2$ are now $d$-twins, so the bubble gadget attached to $S'_1$ collapses to at most one vertex. Vertices $y_1$ and $z$ are now 5-twins, so at most one can survive, which we keep calling $z$. Even if $y_2$ is kept, it is now a $d$-twin of $y_3$, thus at most one of $y_2, y_3$ can be kept. This implies the collapse of the bubble gadget attached to $S'_2$ to at most one vertex, absorbed by $z$. In turn, $y_2$ and $z$ collapse to a single vertex. This process progressively eats up all the vertices $y_j$, and all the bubble gadgets attached to them. As soon as a vertex $x$ or $\neg x$ has three remaining neighbors, it becomes a 6-twin of $z$, and is absorbed by it. We end up with the single vertex $z$.