Problems in NP can Admit Double-Exponential Lower Bounds when Parameterized by Treewidth or Vertex Cover*

Florent Foucaud
Université Clermont Auvergne, CNRS, Mines Saint-Étienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France

Esther Galby
Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden

Liana Khazaliya
Technische Universität Wien, Vienna, Austria

Shaohua Li
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Fionn Mc Inerney
Technische Universität Wien, Vienna, Austria

Roohani Sharma
University of Bergen, Bergen, Norway

Prafullkumar Tale
Indian Institute of Science Education and Research Pune, Pune, India

Abstract

Treewidth serves as an important parameter that, when bounded, yields tractability for a wide class of problems. For example, graph problems expressible in Monadic Second Order (MSO) logic and Quantified SAT or, more generally, Quantified CSP, are fixed-parameter tractable parameterized by the treewidth of the input's (primal) graph plus the length of the MSO-formula [Courcelle, Information & Computation 1990] and the quantifier rank [Chen, ECAI 2004], respectively. The algorithms generated by these (meta-)results have running times whose dependence on treewidth is a tower of exponents. A conditional lower bound by Fichte, Hecher, and Pfandler [LICS 2020] shows that, for Quantified SAT, the height of this tower is equal to the number of quantifier alternations. These types of lower bounds, which show that at least *double-exponential* factors in the running time are *necessary*, exhibit the extraordinary level of computational hardness for such problems, and are rare in the current literature: there are only a handful of such lower bounds (for treewidth and vertex cover parameterizations) and all of them are for problems that are #NP-complete, Σ_2^p-complete, Π_2^p-complete, or complete for even higher levels of the polynomial hierarchy.

Our results demonstrate, for the first time, that it is not necessary to go higher up in the polynomial hierarchy to achieve double-exponential lower bounds: we derive double-exponential lower bounds in the treewidth (tw) and the vertex cover number (vc), for natural, important, and well-studied NP-*complete* graph problems. Specifically, we design a *technique* to obtain such lower bounds and show its *versatility* by applying it to three different problems: Metric Dimension, Strong Metric Dimension, and Geodetic Set. We prove that these problems do not admit $2^{2^{o(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$-time algorithms, even on bounded diameter graphs, unless the ETH fails (here, n is the number of vertices in the graph). In fact, for Strong Metric Dimension, the double-exponential lower bound holds even for the vertex cover number. We further complement all our lower bounds with matching (and sometimes non-trivial) upper bounds.

For the conditional lower bounds, we design and use a novel, yet simple technique based on *Sperner families* of sets. We believe that the amenability of our technique will lead to obtaining

* An extended abstract of parts of this paper was presented in [46].

such lower bounds for many other problems in NP.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Parameterized Complexity, ETH-based Lower Bounds, Double-Exponential Lower Bounds, Kernelization, Vertex Cover, Treewidth, Diameter, Metric Dimension, Strong Metric Dimension, Geodetic Sets

Funding *Florent Foucaud*: ANR project GRALMECO (ANR-21-CE48-0004), French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25), International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25.
Liana Khazaliya: Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029]; Austrian Science Fund (FWF) [Y1329]; European Union's Horizon 2020 COFUND programme [LogiCS@TUWien, grant agreement No. 101034440].
Shaohua Li: European Research Council (ERC) consolidator grant No. 725978 SYSTEMATIC-GRAPH.
Fionn Mc Inerney: Austrian Science Fund (FWF, project Y1329).

Contents

1 Introduction

Many interesting computational problems turn out to be intractable. In these cases, identifying parameters under which the problems become tractable is desirable. In the area of parameterized complexity, treewidth is a cornerstone parameter since a large class of problems become tractable on graphs of bounded treewidth.

Courcelle's celebrated theorem [22] states that the class of graph problems expressible in Monadic Second-Order Logic (MSOL) of constant size is fixed-parameter tractable (FPT) when parameterized by the treewidth of the graph. That is, such problems admit algorithms whose running time is of the form $f(\texttt{tw}) \cdot \mathsf{poly}(n)$, where $\texttt{tw}$ is the treewidth of the input, n is the size of the input, and f is a function that depends only on $\texttt{tw}$. Similarly, a result by Chen [21] shows that the Quantified SAT (Q-SAT) problem can also be solved in time $f(\texttt{tw}) \cdot \mathsf{poly}(n)$, where $\texttt{tw}$ is the treewidth of the primal graph of the input formula and f is a function that depends only on $\texttt{tw}$ and the number of quantifier alternations in the input formula. Q-SAT is a generalization of SAT that allows universal and existential quantifications over the variables. Note that Q-SAT with k quantifier alternations is Π_k^p-complete or Σ_k^p-complete. Unfortunately, in both of the aforementioned results, the function f is a tower of exponents whose height depends roughly on the size of the MSOL and input formulas, respectively. For Q-SAT, the height of this tower equals the number of quantifier alternations in the Q-SAT instance [21].

Over the years, the focus shifted to making such FPT algorithms as efficient as possible. Thus, a natural question is to ask when this higher-exponential dependence on treewidth is necessary. There is a rich literature that provides (conditional) lower bounds on this dependency for many problems, and these bounds are commonly of the form $2^{o(\texttt{tw})}$ or, in some unusual cases, $2^{o(\texttt{tw} \log \texttt{tw})}$ (e.g., [25, 78]) and even $2^{o(\mathsf{poly}(\texttt{tw}))}$ (e.g., [24, 87]). Most notably, these lower bounds are far from the tower of exponents upper bounds given by the (meta-)results discussed above. In this work, we develop a simple technique that allows to prove double-exponential dependence on the treewidth $\texttt{tw}$ and the vertex cover number $\texttt{vc}$, two of the most fundamental graph parameters. Notably, these are the first such results for problems in NP, and we believe that the amenability of our technique will lead to many more similar results for other problems in NP.

Indeed, after a preprint of this paper appeared on arxiv, our technique was also used to prove double-exponential dependence on $\texttt{vc}$ for an NP-complete machine learning problem [19] and double-exponential dependence on the solution size and $\texttt{tw}$ for NP-complete *identification problems* like Test Cover and Locating-Dominating Set [17].

Double-exponential lower bounds: treewidth and vertex cover parameterizations. Fichte, Hecher, and Pfander [42] recently proved that, assuming the Exponential Time Hypothesis[1] (ETH), Q-SAT with k quantifier alternations cannot be solved in time significantly better than a tower of exponents of height k in the treewidth. This exemplifies an interesting but expected trait of this problem: its complexity, in terms of the height of the exponential tower in $\texttt{tw}$, increases with each quantifier alternation. It strengthened the result that appeared in [76], where conditional double-exponential lower bounds for ∃∀SAT and ∀∃SAT were given. The results in [76] also yield a double-exponential lower bound in $\texttt{vc}$ of the primal graph for both problems. Besides these results, there are only a handful of other problems known to require higher-exponential dependence in the treewidth of the input graph (or the

[1] The Exponential Time Hypothesis roughly states that n-variable 3-SAT cannot be solved in time $2^{o(n)}$.

primal graph of the input formula). Specifically, the Π_2^p-complete k-Choosability problem and the Σ_3^p-complete k-Choosability Deletion problem admit a double-exponential and a triple-exponential lower bound in treewidth [80], respectively. Recently, the Σ_p^2-complete problems Cycle HitPack and H-HitPack, for a fixed graph H, were shown to admit tight algorithms that are double-exponential in the treewidth [44]. Further, the Σ_p^2-complete problem Core Stability was shown to admit a tight double-exponential lower bound in the treewidth, even on graphs of bounded degree [53]. Lastly, the #NP-complete counting problem Projected Model Counting admits a double-exponential lower bound in `tw` [40, 41]. For other double-exponential lower bounds, see [1, 26, 45, 53, 61, 69, 71, 72, 75, 79, 88, 94] and Section 3.1.

All the double- (or higher) exponential lower bounds in treewidth mentioned so far are for problems that are #NP-complete, Σ_2^p-complete, Π_2^p-complete, or complete for even higher levels of the polynomial hierarchy. To quote [80]: "Π_2^p*-completeness of these problems already gives sufficient explanation why double- [...] exponential dependence on treewidth is needed. [...] the quantifier alternations in the problem definitions are the common underlying reasons for being in the higher levels of the polynomial hierarchy and for requiring unusually large dependence on treewidth.*"

As mentioned above, we develop a technique that allows to demonstrate, for the first time, that it is not necessary to go to higher levels of the polynomial hierarchy to achieve double-exponential lower bounds in the treewidth or the vertex cover number of the graph.

> We prove that three natural and well-studied NP-complete problems admit double-exponential lower bounds in `tw` or `vc`, under the ETH. These are the first problems in NP known to admit such lower bounds.[2]

NP-complete metric-based graph problems. We study three *metric-based graph problems*. These problems are Metric Dimension, Strong Metric Dimension, and Geodetic Set, and they arise from network design and network monitoring. Apart from serving as examples for double-exponential dependence on treewidth and the amenability of our technique, these problems are of interest in their own right, and possess a rich literature both in the algorithms and discrete mathematics communities (see Section 3.2). Their non-local nature has posed interesting algorithmic challenges and our results, as we explain later, supplement the already vast literature on the structural parameterizations of these problems. Below we define the three above-mentioned problems formally, and particularly focus on Metric Dimension as it is the most popular and well-studied of the three.

> Metric Dimension
> **Input:** A graph G and a positive integer k.
> **Question:** Does there exist $S \subseteq V(G)$ such that $|S| \leq k$ and, for any pair of vertices $u, v \in V(G)$, there exists a vertex $w \in S$ with $d(w,u) \neq d(w,v)$?

The Metric Dimension problem dates back to the 70s [55, 92]. As in geolocation problems, the aim is to distinguish the vertices of a graph via their distances to a solution set. Metric Dimension was first shown to be NP-complete in general graphs in Garey

[2] While it may be possible to artificially engineer a graph problem or graph representation of a problem in NP that admits such lower bounds (although, to the best of our knowledge, this has not been done), we emphasize that this is not the case for these three natural and well-established graph problems in NP.

and Johnson's book [51, GT61], and this was later extended to many restricted graph classes [29, 36, 47], including graphs of diameter 2 [47] and graphs of pathwidth 24 [77]. In a seminal paper, Metric Dimension was proven to be W[2]-hard parameterized by the solution size k, even in subcubic bipartite graphs [56]. This drove the subsequent meticulous study of the problem under structural parameterizations.

In particular, the complexity of Metric Dimension parameterized by treewidth remained an intriguing open problem for a long time. Recently, it was shown that Metric Dimension is para-NP-hard parameterized by pathwidth ($\texttt{pw}$) [77] (an earlier result [12] showed that it is W[1]-hard for pathwidth). A subsequent paper showed that the problem is W[1]-hard parameterized by the combined parameter feedback vertex set number ($\texttt{fvs}$) plus pathwidth of the graph [50]. See Section 3.2 for more related work on Metric Dimension.

We conclude this part with the definitions of the remaining two problems, both of which are known to be NP-Complete [16, 85]. Geodetic Set is also W[1]-hard parameterized by the solution size, feedback vertex set number, and pathwidth, combined [66].

Strong Metric Dimension
Input: A graph G and a positive integer k.
Question: Does there exist $S \subseteq V(G)$ such that $|S| \leq k$ and, for any pair of vertices $u, v \in V(G)$, there exists a vertex $w \in S$ such that either u lies on some shortest path between v and w, or v lies on some shortest path between u and w?

Geodetic Set
Input: A graph G and a positive integer k.
Question: Does there exist $S \subseteq V(G)$ such that $|S| \leq k$ and, for any vertex $u \in V(G)$, there are two vertices $s_1, s_2 \in S$ such that a shortest path from s_1 to s_2 contains u?

Our technical contributions. As Metric Dimension and Geodetic Set are NP-complete on bounded diameter graphs *or* on bounded treewidth graphs, we study their parameterized complexity with $\texttt{tw} + \texttt{diam}$ as the parameter and prove the following results.

1. Metric Dimension and Geodetic Set do not admit algorithms running in time $2^{f(\texttt{diam})^{o(\texttt{tw})}} \cdot n^{\mathcal{O}(1)}$, for any computable function f, unless the ETH fails. (Sections 5, 6)
2. Strong Metric Dimension does not even admit an algorithm with a running time of $2^{2^{o(\texttt{vc})}} \cdot n^{\mathcal{O}(1)}$, unless the ETH fails. This also implies the problem does not admit a kernelization algorithm that outputs an instance with $2^{o(\texttt{vc})}$ *vertices*, unless the ETH fails (Section 7).

The above lower bounds for $\texttt{tw} + \texttt{diam}$, in particular, imply that Metric Dimension and Geodetic Set on graphs of bounded diameter cannot admit $2^{2^{o(\texttt{tw})}} \cdot n^{\mathcal{O}(1)}$-time algorithms, unless the ETH fails. The reduction in Section 5 also works for $\texttt{fvs}$ and $\texttt{td}$ for Metric Dimension, and the reduction in Section 6 works for $\texttt{td}$ for Geodetic Set.

We show that all our lower bounds are tight by providing algorithms (kernelization algorithms, respectively) with matching running times (guarantees, respectively).

1. Metric Dimension and Geodetic Set admit algorithms running in time $2^{\mathtt{diam}^{\mathcal{O}(\mathtt{tw})}} \cdot n^{\mathcal{O}(1)}$. (Sections 8.1, 8.2)
2. Strong Metric Dimension admits an algorithm running in time $2^{2^{\mathcal{O}(\mathtt{vc})}} \cdot n^{\mathcal{O}(1)}$ and a kernel with $2^{\mathcal{O}(\mathtt{vc})}$ vertices. (Section 8.3)

The (kernelization) algorithms for the $\mathtt{vc}$ parameterization are very simple, whereas the algorithms for the $\mathtt{tw} + \mathtt{diam}$ parameter are highly non-trivial and require showing interesting locality properties in the instance. Further, for our $\mathtt{tw} + \mathtt{diam}$ parameterized algorithms, the (double-exponential) dependency of treewidth in the running time is unusual (and rightly so, as exhibited by our lower bounds), as most natural graph problems in NP for which a dedicated algorithm (i.e., *not* relying on Courcelle's theorem) parameterized by treewidth is known, can be solved in time $2^{\mathcal{O}(\mathtt{tw})} \cdot n^{\mathcal{O}(1)}$, $2^{\mathcal{O}(\mathtt{tw} \cdot \log(\mathtt{tw}))} \cdot n^{\mathcal{O}(1)}$ or $2^{\mathcal{O}(\mathsf{poly}(\mathtt{tw}))} \cdot n^{\mathcal{O}(1)}$.

Finally, our reductions rely on a novel, yet simple technique based on *Sperner families* of sets that allows to encode particular SAT relations across large sets of variables and clauses into relatively small vertex-separators. As mentioned before, we believe that this technique is the key to obtaining such lower bound results for other problems in NP. In particular, as witnessed by our results, our technique has the additional features that it even allows to prove such lower bounds in very restricted cases, such as bounded diameter graphs, and is not specific to any one structural parameter, as it also works for, e.g., the feedback vertex set number and treedepth. We elaborate on our technique in the next section.

2 Technical Overview

In this section, we present an overview of our lower bound techniques. We first exhibit our technique to obtain the double-exponential lower bounds in its most general setting. Then, we continue with the problem-specific tools we developed that are required for the reductions.

The first integral part of our technique is to reduce from a variant of 3-SAT known as 3-Partitioned-3-SAT that was introduced in [74]. In this problem, the input is a formula ψ in 3-CNF form, together with a partition of the set of its variables into three disjoint sets X^α, X^β, X^γ, with $|X^\alpha| = |X^\beta| = |X^\gamma| = n$, and such that no clause contains more than one variable from each of X^α, X^β, and X^γ. The objective is to determine whether ψ is satisfiable. Unless the ETH fails, 3-Partitioned-3-SAT does not admit an algorithm running in time $2^{o(n)}$ [74, Theorem 3].

Typical reductions from satisfiability problems to graph problems usually entail representing the satisfiability problem by its incidence graph, in which each variable is represented by two vertices corresponding to its positive and negative literals. In this representation, a clause vertex is adjacent to a literal vertex if and only if it contains that literal in ψ (see Figure 1 (left) for an illustration). However, this naive approach does not lead to any structural parameters of the incidence graph being of bounded size. The *core idea* of our technique is to instead represent the relationships between clause and literal vertices via edges from these two sets of vertices to "small" separators (three separators in the case of 3-Partitioned-3-SAT) that encode these relationships.

Formally, this is achieved as follows. For a positive integer p, define $\mathcal{F}_p$ as the collection of subsets of $[2p]$ that contains exactly p integers. We critically use the fact that no set in $\mathcal{F}_p$ is contained in any other set in $\mathcal{F}_p$ (such a collection of sets are called a *Sperner family*). Let ℓ be a positive integer such that $\ell \leq \binom{2p}{p}$. We define $\mathsf{set\text{-}rep} : [\ell] \mapsto \mathcal{F}_p$ as a one-to-one function by arbitrarily assigning a set in $\mathcal{F}_p$ to an integer in $[\ell]$. By the asymptotic estimation of the

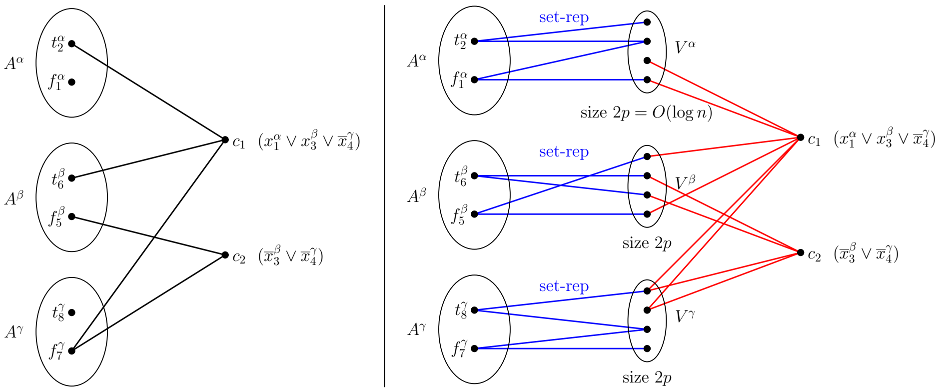

Figure 1 Graph representations of 3-Partitioned-3-SAT. (Left) incidence graph representation. (Right) representation with small separators using our technique. Note, for example, that x_1^α appears as a positive literal in the clause C_1. Thus, on the left, t_2^α is the only literal vertex in A^α incident to c_1, while on the right, t_2^α is the only literal vertex in A^α that does not share a common neighbor with c_1 in V^α. The edges from c_2 to each vertex in V^α are omitted for clarity.

central binomial coefficient, $\binom{2p}{p} \sim \frac{4^p}{\sqrt{\pi \cdot p}}$ [59]. To get the upper bound of p, we scale down the asymptotic function and have $\ell \leq \frac{4^p}{2^p} = 2^p$. Thus, $p = \mathcal{O}(\log \ell)$.

Let ψ be an instance of 3-Partitioned-3-SAT on $3n$ variables, and let p be the smallest integer such that $2n \leq \binom{2p}{p}$. In particular, $p = \mathcal{O}(\log n)$. Define $\mathsf{set\text{-}rep} : [2n] \mapsto \mathcal{F}_p$ as above. Rename the variables in X^α to x_i^α for all $i \in [n]$. For each variable x_i^α, add two vertices t_{2i}^α and f_{2i-1}^α corresponding to the positive and negative literals of x_i^α, respectively. Let $A^\alpha = \{t_{2i}^\alpha, f_{2i-1}^\alpha |\ i \in [n]\}$. Add a *validation portal* with $2p$ vertices, denoted by $V^\alpha = \{v_1^\alpha, \ldots, v_{2p}^\alpha\}$. For each $i \in [n]$, add the edge $t_{2i}^\alpha v_{p'}^\alpha$ for each $p' \in \mathsf{set\text{-}rep}(2i)$. Similarly, for each $i \in [n]$, add the edge $f_{2i-1}^\alpha v_{p'}^\alpha$ for each $p' \in \mathsf{set\text{-}rep}(2i-1)$. Repeat the above steps for β and γ.

Now, for each clause C_j ($j \in [m]$) in ψ, add a clause vertex c_j. Let $\delta \in \{\alpha, \beta, \gamma\}$. For all $i \in [n]$ and $j \in [m]$, if the variable x_i^δ appears as a positive (negative, respectively) literal in the clause C_j in ψ, then add the edge $c_j v_{p'}^\delta$ for each $p' \in [2p] \setminus \mathsf{set\text{-}rep}(2i)$ ($p' \in [2p] \setminus \mathsf{set\text{-}rep}(2i-1)$, respectively). For all $j \in [m]$, if no variable from X^δ appears in C_j in ψ, then make c_j adjacent to all the vertices in V^δ. See Figure 1 (right) for an illustration.

As a clause contains at most one variable from X^δ in ψ, c_j and t_{2i}^δ (f_{2i-1}^δ, respectively) do not share a common neighbor in V^δ if and only if the clause C_j contains x_i^δ as a positive (negative, respectively) literal in ψ. For the reductions, we use this representation of the relationship between clause and literal vertices. Since $p = \mathcal{O}(\log n)$, this ensures that $\mathtt{tw}(G) = \mathcal{O}(\log n)$, which we exploit along with the fact that, unless the ETH fails, 3-Partitioned-3-SAT does not admit an algorithm running in time $2^{o(n)}$.

2.1 Basic Tools for Lower Bounds

For brevity, we focus on Metric Dimension and explain our problem-specific tools in this context. We use two such simple tools: the *bit representation gadget* and the *set representation gadget*. The set representation gadget is the problem-specific implementation of the above technique, and it uses the bit representation gadget.

Before going further, we need to define some terms related to Metric Dimension. The set S defined in the problem statement of Metric Dimension is called a *resolving set* of G. A subset of vertices $S' \subseteq V(G)$ *resolves* a pair of vertices $u, v \in V(G)$ if there exists a vertex $w \in S'$ such that $d(w, u) \neq d(w, v)$. Lastly, a vertex $u \in V(G)$ is *distinguished* by a

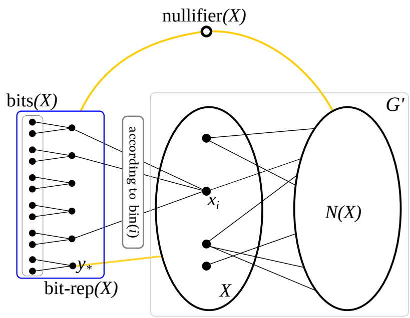

Figure 2 Set Identifying Gadget. The blue box represents $\mathsf{bit\text{-}rep}(X)$ and the yellow lines represent that $\mathsf{nullifier}(X)$ is adjacent to each vertex in $\mathsf{bit\text{-}rep}(X) \cup N(X)$, and $y_\star$ is adjacent to each vertex in X. Also, G' is not necessarily restricted to the graph induced by the vertices in $X \cup N(X)$.

subset of vertices $S' \subseteq V(G)$ if, for any $v \in V(G) \setminus \{u\}$, there exists a vertex $w \in S'$ such that $d(w,u) \neq d(w,v)$.

Bit Representation Gadget to Identify Sets. Suppose we are given a graph G' and a subset $X \subseteq V(G')$ of its vertices. Further, suppose that we want to add a vertex set X^+ to G' to obtain a new graph G with the following properties. We want that each vertex in $X \cup X^+$ is distinguished by vertices in X^+ that must be in any resolving set S of G, and that no vertex in X^+ can resolve any "critical pair" of vertices in G. Roughly, a pair of vertices is critical if it forces certain "types" of vertices to be in any resolving set S of G, and the selection of the specific vertices of those types depends on the solution to the problem being reduced from (which, in our case, is 3-Partitioned-3-SAT [74]). We refer to the graph induced by the vertices of X^+, along with the edges connecting X^+ to G', as the *Set Identifying Gadget* for the set X. Given a graph G' and a non-empty subset $X \subseteq V(G')$ of its vertices, to construct such a graph G, we add vertices and edges to G' as follows (see Figure 2):

- The vertex set X^+ that we are aiming to add is the union of a set $\mathsf{bit\text{-}rep}(X)$ and a special vertex denoted by $\mathsf{nullifier}(X)$.
- First, let $X = \{x_i \mid i \in [|X|]\}$, and set $q := \lceil \log(|X|+2) \rceil + 1$. We select this value for q to *(1)* uniquely represent each integer in $[|X|]$ by its bit-representation in binary (note that we start from 1 and not 0), *(2)* ensure that the only vertex whose bit-representation contains all 1's is $\mathsf{nullifier}(X)$, and *(3)* reserve one spot for an additional vertex $y_\star$.
- For every $i \in [q]$, add three vertices y_i^a, y_i, y_i^b, and add the path (y_i^a, y_i, y_i^b).
- Add 3 vertices $y_\star^a, y_\star, y_\star^b$ and the path $(y_\star^a, y_\star, y_\star^b)$. Add edges to make $\{y_i \mid i \in [q]\} \cup \{y_\star\}$ a clique. Make $y_\star$ adjacent to each vertex in X. Let $\mathsf{bit\text{-}rep}(X) = \{y_i, y_i^a, y_i^b \mid i \in [q]\} \cup \{y_\star, y_\star^a, y_\star^b\}$ and denote its subset by $\mathsf{bits}(X) = \{y_i^a, y_i^b \mid i \in [q]\} \cup \{y_\star^a, y_\star^b\}$.
- For every integer $j \in [|X|]$, let $\mathsf{bin}(j)$ denote the binary representation of j using q bits. Connect x_j with y_i if the i^{th} bit (going from left to right) in $\mathsf{bin}(j)$ is 1.
- Add a vertex, denoted by $\mathsf{nullifier}(X)$, and connect it to each vertex in $\{y_i \mid i \in [q]\} \cup \{y_\star\}$.
- For every vertex $u \in V(G) \setminus (X \cup X^+)$ such that u is adjacent to some vertex in X, add an edge between u and $\mathsf{nullifier}(X)$. We add this vertex to ensure that vertices in $\mathsf{bit\text{-}rep}(X)$ do not resolve critical pairs in $V(G)$.

Set Representation Gadget. We define $\mathsf{set\text{-}rep} : [\ell] \mapsto \mathcal{F}_p$ as in Section 2, and recall that $p = \mathcal{O}(\log \ell)$. Suppose we have a "large" collection of vertices, say $A = \{a_1, a_2, \dots, a_\ell\}$, and a "large" collection of critical pairs $C = \{\langle c_1^\circ, c_1^\star \rangle, \langle c_2^\circ, c_2^\star \rangle, \dots, \langle c_m^\circ, c_m^\star \rangle\}$. Moreover, we are

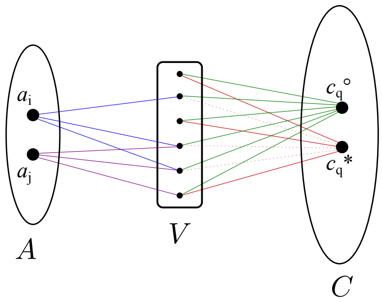

Figure 3 Set Representation Gadget. Let $\phi(q) = i$, i.e., only a_i in A can resolve the critical pair $\langle c_q^\circ, c_q^\star \rangle$. Let the vertices in V be indexed from top to bottom and let $\textsf{set-rep}(i) = \{2, 4, 5\}$. By construction, the only vertices in V that $c_q^\star$ is *not* adjacent to are v_2, v_4, and v_5 (this is highlighted by red-dotted edges). Thus, $\text{dist}(a_i, c_q^\circ) = 2$ and $\text{dist}(a_i, c_q^\star) > 2$, and hence, a_i resolves $\langle c_q^\circ, c_q^\star \rangle$. For any other vertex in A, say a_j, $\textsf{set-rep}(j) \setminus \textsf{set-rep}(i)$ is non-empty, and thus, a_j cannot resolve $\langle c_q^\circ, c_q^\star \rangle$.

given an injective function $\phi : [m] \mapsto [\ell]$. The objective is to design a gadget such that only $a_{\phi(q)} \in A$ can resolve a critical pair $\langle c_q^\circ, c_q^\star \rangle \in C$ for any $q \in [m]$, while keeping the treewidth of this part of the graph of order $\mathcal{O}(\log(|A|))$. With this in mind, we do the following.

- Add vertices and edges to identify the set A and to add critical pairs in C (for each critical pair in C, both vertices share the same bit-representation in the Set Identifying Gadget for C).
- Add a *validation portal*, a clique on $2p$ vertices, denoted by $V = \{v_1, v_2, \ldots, v_{2p}\}$, and vertices and edges to identify it.
- For every $i \in [\ell]$ and for every $p' \in \textsf{set-rep}(i)$, add the edge $(a_i, v_{p'})$.
- For every critical pair $\langle c_q^\circ, c_q^\star \rangle$, make c_q° adjacent to every vertex in V, and add every edge of the form $(c_q^\star, v_{p'})$ for $p' \in [2p] \setminus \textsf{set-rep}(\phi(q))$. Note that the vertices in V that are indexed using integers in $\textsf{set-rep}(\phi(q))$ are *not* adjacent with $c_q^\star$.

See Figure 3 for an illustration. Now, consider a critical pair $\langle c_q^\circ, c_q^\star \rangle$ and suppose $i = \phi(q)$.

- By the construction, $N(a_i) \cap N(c_q^\circ) \neq \emptyset$, whereas $N(a_i) \cap N(c_q^\star) = \emptyset$. Hence, a_i resolves the critical pair $\langle c_q^\circ, c_q^\star \rangle$ as $d(a_i, c_q^\circ) = 2$ and $d(a_j, c_q^\star) > 2$.
- For any other vertex in A, say a_j, $\textsf{set-rep}(j) \setminus \textsf{set-rep}(i)$ is a non-empty set. So, there are paths from a_j to c_q° and a_j to $c_q^\star$ through vertices in V with indices in $\textsf{set-rep}(j) \setminus \textsf{set-rep}(i)$. This implies that $d(a_j, c_q^\circ) = d(a_j, c_q^\star) = 2$ and a_j cannot resolve the pair $\langle c_q^\circ, c_q^\star \rangle$.

2.2 Sketch of the Lower Bound Proof for Metric Dimension

With these tools in hand, we present an overview of the reduction from 3-Partitioned-3-SAT used to prove Theorem 6, which we restate here for convenience.

Theorem 6. *Unless the ETH fails, Metric Dimension does not admit an algorithm running in time* $2^{f(\texttt{diam})^{o(\texttt{tw})}} \cdot n^{\mathcal{O}(1)}$ *for any computable function* $f : \mathbb{N} \mapsto \mathbb{N}$.

The reduction in the proof of Theorem 6 takes as input an instance ψ of 3-Partitioned-3-SAT on $3n$ variables and returns (G, k) as an instance of Metric Dimension such that $\texttt{tw}(G) = \mathcal{O}(\log(n))$ and $\texttt{diam}(G) = \mathcal{O}(1)$. In the following, we mention a crude outline of the reduction, omitting some technical details. For the formal proof, please refer to Section 5.

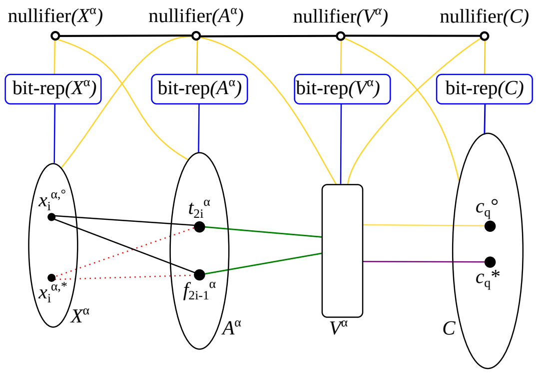

Figure 4 Reduction for proof of Theorem 6. Yellow lines represent that vertex is connected to every vertex in the set the edge goes to. Green edges denote adjacencies with respect to set-rep, e.g., t_{2i}^{α} is adjacent to $v_j \in V^\alpha$ if $j \in$ set-rep$(2i)$. Purple lines also indicate adjacencies with respect to set-rep, but in a complementary way, i.e., if $x_i \in c_q$, then, for every $p' \in [2p] \setminus$ set-rep$(2i)$, we have $(v_{p'}^{\alpha}, c_q^{\star}) \in E(G)$, and if $\overline{x}_i \in c_q$, then, for all $p' \in [2p] \setminus$ set-rep$(2i-1)$, we have $(v_{p'}^{\alpha}, c_q^{\star}) \in E(G)$.

2.2.1 Reduction

- We rename the variables in X^α to x_i^α for $i \in [n]$. For every variable x_i^α, we add a critical pair $\langle x_i^{\alpha,\circ}, x_i^{\alpha,\star}\rangle$ of vertices. We denote $X^\alpha = \{x_i^{\alpha,\circ}, x_i^{\alpha,\star} \mid i \in [n]\}$.
- For each variable x_i^α, we add the vertices t_{2i}^α and f_{2i-1}^α. Let $A^\alpha = \{t_{2i}^\alpha, f_{2i-1}^\alpha | \ i \in [n]\}$.
- For every $i \in [n]$, we add the edges $(x_i^{\alpha,\circ}, t_{2i}^\alpha)$ and $(x_i^{\alpha,\circ}, f_{2i-1}^\alpha)$ which will ensure that any resolving set contains at least one vertex in $\{t_{2i}^\alpha, f_{2i-1}^\alpha, x_i^{\alpha,\circ}, x_i^{\alpha,\star}\}$ for every $i \in [n]$.
- Let p be the smallest integer such that $2n \le \binom{2p}{p}$. In particular, $p = \mathcal{O}(\log n)$. Define set-rep : $[2n] \mapsto \mathcal{F}_p$ as in Section 2.
- We add a *validation portal*, a clique on $2p$ vertices, denoted by $V^\alpha = \{v_1^\alpha, v_2^\alpha, \dots, v_{2p}^\alpha\}$.
- For each $i \in [n]$, we add the edge $(t_{2i}^\alpha, v_{p'}^\alpha)$ for every $p' \in$ set-rep$(2i)$. Similarly, for each $i \in [n]$, we add the edge $(f_{2i-1}^\alpha, v_{p'}^\alpha)$ for every $p' \in$ set-rep$(2i-1)$.

  We repeat the above steps to construct $X^\beta, A^\beta, V^\beta$, $X^\gamma, A^\gamma, V^\gamma$.

- For every clause C_q in ψ, we introduce a pair $\langle c_q^\circ, c_q^\star\rangle$ of vertices. Let C be the collection of vertices in such pairs.
- We add edges across C and the portals as follows. Consider a clause C_q in ψ and the corresponding critical pair $\langle c_q^\circ, c_q^\star\rangle$ in C. Let $\delta \in \{\alpha, \beta, \gamma\}$. As ψ is an instance of 3-Partitioned-3-SAT, at most one variable in X^δ appears in C_q, say x_i^δ for some $i \in [n]$. We add all edges of the form $(v_{p'}^\delta, c_q^\circ)$ for every $p' \in [2p]$. If x_i^δ appears as a positive literal in C_q, then we add the edge $(v_{p'}^\delta, c_q^\star)$ for every $p' \in [2p] \setminus$ set-rep$(2i)$ (which corresponds to t_{2i}^δ). If x_i^δ appears as a negative literal in C_q, then we add the edge $(v_{p'}^\delta, c_q^\star)$ for every $p' \in [2p] \setminus$ set-rep$(2i-1)$ (which corresponds to f_{2i-1}^δ). Note that if x_i^δ appears as a positive (negative, respectively) literal in C_q, then the vertices in V^δ whose indices are in set-rep$(2i)$ (set-rep$(2i-1)$, respectively) are *not adjacent* to $c_q^\star$. If no variable in X^δ appears in C_q, then we make each vertex in V^δ adjacent to both c_q° and $c_q^\star$.

For all the sets mentioned above, we add vertices and edges to identify them as shown in Figure 4 (for each critical pair, both vertices share the same bit-representation in their Set Identifying Gadget). This concludes the construction of G. The reduction returns (G, k) as an instance of Metric Dimension for some appropriate value of k.

2.2.2 Correctness of the Reduction

We give an informal description of the proof of correctness of the reverse direction here. Fix $\delta \in \{\alpha, \beta, \gamma\}$. For all $i \in [n]$, the only vertices that can resolve the critical pair $\langle x_i^{\delta,\circ}, x_i^{\delta,\star}\rangle$ are the vertices in $\{x_i^{\delta,\circ}, x_i^{\delta,\star}\} \cup \{t_{2i}^{\delta}, f_{2i-1}^{\delta}\}$. This fact and the budget k ensure that any resolving set of G contains exactly one vertex from $\{t_{2i}^{\delta}, f_{2i-1}^{\delta}\} \cup \{x_i^{\delta,\circ}, x_i^{\delta,\star}\}$ for all $i \in [n]$. This naturally corresponds to an assignment of the variable x_i^{δ} if a vertex from $\{t_{2i}^{\delta}, f_{2i-1}^{\delta}\}$ is in the resolving set. However, if a vertex from $\{x_i^{\delta,\circ}, x_i^{\delta,\star}\}$ is in the resolving set, then we can see this as giving an arbitrary assignment to the variable x_i^{δ}. Suppose the clause C_q contains the variable x_i^{δ} as a positive literal. By the construction, every vertex in V^{δ} that is adjacent to t_{2i}^{δ} is *not* adjacent to $c_q^{\star}$. However, c_q° is adjacent to every vertex in V^{δ}. Hence, $d(t_{2i}^{\delta}, c_q^{\circ}) = 2$, whereas $d(t_{2i}^{\delta}, c_q^{\star}) > 2$. Thus, t_{2i}^{δ} resolves the critical pair $\langle c_q^{\circ}, c_q^{\star}\rangle$. Consider any other vertex in A^{δ}, say t_{2j}^{δ}. Since $\textsf{set-rep}(2i)$ is not a subset of $\textsf{set-rep}(2j)$ (as both have the same cardinality), there is at least one integer, say p', in $\textsf{set-rep}(2j) \setminus \textsf{set-rep}(2i)$. The vertex $v_{p'}^{\delta} \in V^{\delta}$ is adjacent to t_{2j}^{δ}, c_q°, and $c_q^{\star}$. Hence, t_{2j}^{δ} cannot resolve the critical pair $\langle c_q^{\circ}, c_q^{\star}\rangle$ as both these vertices are at distance 2 from it. Also, as ψ is an instance of 3-Partitioned-3-SAT, C_q contains at most one variable in X^{δ}, which is x_i^{δ} in this case. This also helps to encode the fact that at most one vertex from A^{δ} should be able to resolve the critical pair $\langle c_q^{\circ}, c_q^{\star}\rangle$. Since vertices in X^{δ} cannot resolve critical pairs $\langle c_q^{\circ}, c_q^{\star}\rangle$ in C, then finding a resolving set in G corresponds to finding a satisfying assignment for ψ.

2.2.3 Lower Bounds Obtained from the Reduction

Let $Z = \{V^{\delta} \cup X^{+} \mid X \in \{X^{\delta}, A^{\delta}, V^{\delta}, C\}, \delta \in \{\alpha, \beta, \gamma\}\}$. Note that $|Z| = \mathcal{O}(\log(n))$ and $G - Z$ is a collection of P_3's and isolated vertices. Hence, $\texttt{tw}(G)$, $\texttt{fvs}(G)$, and $\texttt{td}(G)$ are upper bounded by $\mathcal{O}(\log(n))$. Also, G has constant diameter. Thus, if there is an algorithm for Metric Dimension that runs in time $2^{f(\texttt{diam})^{o(\texttt{tw})}}$ (or $2^{f(\texttt{diam})^{o(\texttt{fvs})}}$ or $2^{f(\texttt{diam})^{o(td)}}$), then there is an algorithm solving 3-Partitioned-3-SAT in time $2^{o(n)}$, contradicting the ETH.

3 Related Work

3.1 Double-Exponential Lower Bounds

It is long known that certain algorithmic tasks cannot be solved in less than double-exponential time. In the realm of classical complexity, this is captured by the complexity class 2-EXPTIME, with some problems that are complete for that class, for example, Presburger Arithmetic [43], the Asynchronous Reactive Module Synthesis arising from linear temporal logic [89] or Planning with Partial Observability [90].

In the realm of parameterized complexity as well, it is known that certain problems have a fixed-parameter-tractable running time that requires a double-exponential dependency in the parameter, such as Edge Clique Cover [26], Multi-Team Formation [79], Modal Satisfiability [1], and Distinct Vectors [88].

When it comes to structural parameterized algorithms, treewidth is one of the main success stories of the field, as many problems are FPT when parameterized by the treewidth of the input. This is witnessed by the famous theorem of Courcelle that states that all problems expressible in MSOL can be solved by a linear-time FPT algorithm [22], and by a similar result of Chen for the Quantified SAT and Quantified CSP problems [21] (here, we consider the treewidth of the primal graph of the input relational structure). Unfortunately, the dependence in treewidth is notoriously “galactic”: a tower of exponentials whose height depends on the number of quantifier alternations in the MSOL formula, and in the SAT instance, respectively. Moreover, Chen [21] showed that the height of this tower is equal to the number of quantifier alternations in the Quantified SAT instance.

However, note that Quantified SAT and MSO Model Checking on Trees (even for FO formulas) are PSPACE-complete problems [3, 48]. There are few natural problems that have been shown to admit (at least) double-exponential lower bounds with respect to treewidth. The Π_2^p-complete k-Choosability problem and the Σ_3^p-complete k-Choosability Deletion problem admit a double-exponential and a triple-exponential lower bound in treewidth [80], respectively. Double-exponential lower bounds for the Σ_2^p-complete $\exists\forall$SAT and Π_2^p-complete $\forall\exists$SAT problems were shown in [76]. Recently, the Σ_p^2-complete problems Cycle HitPack and H-HitPack, for a fixed graph H, were shown to admit tight algorithms that are double-exponential in the treewidth [44]. Further, the Σ_p^2-complete problem Core Stability was shown to admit a tight double-exponential lower bound in the treewidth, even on graphs of bounded degree [53]. Lastly, the #NP-complete counting problem Projected Model Counting admits a double-exponential lower bound in tw [40, 41]. Similar lower bounds were obtained in [75] for problems from artificial intelligence (abstract argumentation, abduction, circumscription, and the computation of minimal unsatisfiable sets in unsatisfiable formulas), with these problems also lying in the second level of the polynomial hierarchy.

With respect to the vertex cover parameter (of the primal graph of the input relational structure), the Σ_2^p-complete problem $\exists\forall$-CSP also requires a double-exponential dependency in its running time [76].

A double-exponential lower bound is also known for Coloring with respect to the smaller parameter *cliquewidth* [45]. However, in contrast with the aforementioned ones, this problem is only XP and not FPT parameterized by the cliquewidth. As another example, we know that ILP Feasibility admits a double-exponential lower bound when parameterized by the *dual treedepth* of the input matrix [69].

3.2 Metric Graph Problems

Metric graph problems are defined using either distance values or shortest paths in the graph. Metric-based graph problems are ubiquitous in computer science, for example, the classic (Single-Source) Shortest Path, (Graphic) Traveling Salesperson Problem or Steiner Tree fall into this category. Those are fundamental problems, often stemming from applications in network design, for which a lot of algorithmic research has been done. Among these, metric-based graph packing and covering problems such as, for example, Distance Domination [60] or Scattered Set [65], have recently gained a lot of attention. Their non-local nature leads to non-trivial algorithmic properties that differ from most classic graph problems with a more local nature. This is the case in particular for treewidth-based algorithms. In this paper, we focus on three problems arising from network design (Geodetic Set) and network monitoring (Metric Dimension and Strong Metric Dimension). These problems have far-reaching applications, as exemplified by, e.g., the recent work [11], in which it was shown that enumerating minimal solution sets for the metric dimension problem in (general) graphs and the geodetic set problem in split graphs is equivalent to enumerating minimal transversals of hypergraphs, arguably the most important open problem in algorithmic enumeration.

Metric Dimension. Metric Dimension was introduced in the 1970s independently by Harary and Melter [55] and Slater [92] as a network monitoring problem. Metric Dimension and its variants (see, e.g., [9, 10, 37, 49, 54, 64, 91, 93]) are very well-studied and have numerous applications such as in graph isomorphism testing [5], network discovery [7], image processing [82], chemistry [62], graph reconstruction [81] or genomics [95]. In fact, Metric Dimension was first shown to be NP-complete in general graphs in Garey and Johnson's book [51], and this was later extended to unit disk graphs [58], split graphs, bipartite graphs, co-bipartite graphs, and line graphs of bipartite graphs [36], bounded-degree planar graphs [29], and interval and permutation graphs of diameter 2 [47]. On the tractable side, Metric Dimension admits linear-time algorithms on trees [92], cographs [36], chain graphs [39], cactus block graphs [57], and bipartite distance-hereditary graphs [84], and a polynomial-time algorithm on outerplanar graphs [29].

Due to the NP-hardness results, the focus has now shifted to studying its parameterized complexity, in search of tractable instances. In a seminal paper, it was proven that Metric Dimension is W[2]-hard parameterized by the solution size k, even in subcubic bipartite graphs [56]. This paper was the driving motivation behind the subsequent meticulous study of Metric Dimension under structural parameterizations. Several different parameterizations have been studied for this problem, that we now elaborate on (see also [50, Figure 1]).

In terms of structural parameterizions for Metric Dimension, through careful design, kernelization, and/or meta-results, it was proven that there is an XP algorithm parameterized by the feedback edge set number in [36], and FPT algorithms parameterized by the max leaf number in [35], the modular-width and the treelength plus the maximum degree in [8], the treedepth and the clique-width plus the diameter in [52], and the distance to cluster (co-cluster, respectively) in [50]. Recently, an FPT algorithm parameterized by the treewidth in chordal graphs was given in [13]. On the negative side, Metric Dimension is W[1]-hard parameterized by the pathwidth on graphs of constant degree [12], para-NP-hard parameterized by the pathwidth [77], and W[1]-hard parameterized by the combined parameter feedback vertex set number plus pathwidth [50]. Lastly, it is not computable (unless the ETH fails) in time $2^{o(n)}$ on bipartite graphs, and in time $2^{o(\sqrt{n})}$ on planar bipartite graphs [6].

Strong Metric Dimension. Albeit less well-studied than Metric Dimension, the Strong Metric Dimension problem, which was introduced by Sebő and Tannier in 2004 [91] as a strengthening of Metric Dimension, enjoys interesting applications in coin-weighing problems and other areas of algorithms and combinatorics. Here, we are given a graph G and an integer k, and we are seeking a solution set S of size at most k, called a *strong resolving set*, such that, for any pair of vertices $u, v \in V(G)$, there exists a vertex $w \in S$ such that either u lies on some shortest path between v and w, or v lies on some shortest path between u and w. The size of the smallest such set S is called the *strong metric dimension* of G. It is clear from the definition that any strong resolving set is also a resolving set.

In the seminal paper introducing the problem, it was used to design an efficient algorithm for the graph problem Connected Join Existence [91]. Interestingly, it was shown in [85] that the problem on an instance (G, k) can be reduced (in polynomial time) to an instance (G', k) of Vertex Cover where $V(G) = V(G')$ and the edges of G' join a set of suitably defined critical pairs (see also [73] for further studies of this reduction). Consequently, algorithmic results known for Vertex Cover can be applied to Strong Metric Dimension: in particular, the problem is FPT when parameterized by the solution size. On the other hand, it was shown in [27] that many hardness results known for Vertex Cover can also be transferred to Strong Metric Dimension.

Geodetic Set. Geodetic Set was introduced in 1993 by Harary, Loukakis, and Tsouros in [54]. It can be seen as a network design problem, where one seeks to determine the optimal locations of public transportation hubs in a road network, while minimizing the total number of such hubs [16]. Other applications are mentioned in [34]. More generally, Geodetic Set is part of the area of geodesic convexity in graphs: see, e.g., the paper [38] or the book [86].

As is often the case with metric-based problems, Geodetic Set is computationally hard, even for very structured graphs. Its NP-hardness was claimed in the seminal paper [54] (see [33] for the earliest explicit proofs). This is known to hold even for graphs that belong to various structured input graph classes, such as interval graphs [15], co-bipartite graphs [34], line graphs [16], graphs of diameter 2 [16], and subcubic (planar bipartite) grid graphs of arbitrarily large girth [15, 18] (see also [4, 14, 31, 32] for various earlier hardness results). Geodetic Set can be solved in polynomial time on split graphs [32, 33] and, more generally, well-partitioned chordal graphs [2], outerplanar graphs [83], ptolemaic graphs [38], cographs [32] and, more generally, distance-hereditary graphs [63], block-cactus graphs [34], solid grid graphs [15, 18], and proper interval graphs [34].

The parameterized complexity of Geodetic Set was first addressed by Kellerhals and Koana in [66]. They observed that the reduction from [32] implies that the problem is W[2]-hard when parameterized by the solution size (even for chordal bipartite graphs). The above-mentioned hardness results for structural graph classes motivated the authors of [66] to investigate structural parameterizations of Geodetic Set. They proved the problem to be W[1]-hard for the parameters solution size, feedback vertex set number, and pathwidth, combined [66]. On the positive side, they showed that Geodetic Set is FPT for the parameters treedepth, modular-width (more generally, clique-width plus diameter), and feedback edge set number [66]. The problem is also FPT on chordal graphs when parameterized by the treewidth [15].

The approximability of Geodetic Set was also studied. Its minimization variant is NP-hard to approximate within a factor of $o(\log n)$, even for diameter 2 graphs [16] and subcubic bipartite graphs of arbitrarily large girth [28]. It can be approximated in polynomial time within a factor of $n^{1/3} \log n$ [16] (but the best possible approximation factor is unknown).

4 Preliminaries

In this paper, all logarithms are to the base 2. For an integer a, we let $[a] = \{1, \ldots, a\}$.

Graph theory. We use standard graph-theoretic notation and refer the reader to [30] for any undefined notation. For an undirected graph G, the sets $V(G)$ and $E(G)$ denote its set of vertices and edges, respectively. Two vertices $u, v \in V(G)$ are *adjacent* or *neighbors* if $(u, v) \in E(G)$. The *open neighborhood* of a vertex $u \in V(G)$, denoted by $N(u) := N_G(u)$, is the set of vertices that are neighbors of u. The *closed neighborhood* of a vertex $u \in V(G)$ is denoted by $N[u] := N_G[u] := N_G(u) \cup \{u\}$. For any $X \subseteq V(G)$ and $u \in V(G)$, $N_X(u) = N_G(u) \cap X$. Any two vertices $u, v \in V(G)$ are *true twins* if $N[u] = N[v]$, and are *false twins* if $N(u) = N(v)$. Observe that if u and v are true twins, then $(u, v) \in E(G)$, but if they are only false twins, then $(u, v) \notin E(G)$. For a subset S of $V(G)$, we say that the vertices in S are true (false, respectively) twins if, for any $u, v \in S$, u and v are true (false, respectively) twins. The *distance* between two vertices $u, v \in V(G)$ in G, denoted by $d(u, v) := d_G(u, v)$, is the length of a (u, v)-shortest path in G. For a subset S of $V(G)$, we define $N[S] = \bigcup_{v \in S} N[v]$ and $N(S) = N[S] \setminus S$. For a subset S of $V(G)$, we denote the graph obtained by deleting S from G by $G - S$. We denote the subgraph of G induced on the set S by $G[S]$. For a graph G, a set $X \subseteq V(G)$ is a *vertex cover* of G if $V(G) \setminus X$ is an independent set. We denote by $\texttt{vc}(G)$ the size of a minimum vertex cover in G. When G is clear from the context, we simply say vc. For a graph G, a set $X \subseteq V(G)$ is a *feedback vertex set* of G if $V(G) \setminus X$ is an acyclic graph. We define the notation of *the feedback vertex set number* in the analogous way.

Tree decompositions. A *tree decomposition* of a graph G is a pair $(T, \mathcal{X})$, where T is a tree and $\mathcal{X} := \{X_i : i \in V(T)\}$ is a collection of subsets of $V(G)$, called *bags*, satisfying the following conditions: (i) $\bigcup_{i \in V(T)} X_i = V(G)$, (ii) for every edge $(u, v) \in E(G)$, there is a bag that contains both u and v, and (iii) for every vertex $v \in V(G)$, the set of nodes of T whose bags contain v induces a (connected) subtree of T.

The maximum size of a bag minus one is called the *width* of T. The minimum width of a tree decomposition of G is the *treewidth* of G.

We consider a *rooted* tree decomposition by fixing a root of T and orienting the tree edges from the root toward the leaves. A rooted tree decomposition is *nice* (see [68]) if each node i of T has at most two children and falls into one of the four types:

- *Join* node: i has exactly two children i_1 and i_2 with $X_i = X_{i_1} = X_{i_2}$.
- *Introduce* node: i has a unique child i' with $X_{i'} = X_i \setminus \{v\}$, where $v \in V(G) \setminus X_{i'}$.
- *Forget* node: i has a unique child i' with $X_i = X_{i'} \setminus \{v\}$, where $v \in X_{i'}$.
- *Leaf* node: i is a leaf of T with $|X_i| = 1$.

For a node i of T, we denote by T_i the subtree of T rooted at i, and by G_i, the subgraph of G induced by the vertices of the bags in T_i.

For a graph G, a set $S \subseteq V(G)$ is a *separator* for two non-adjacent vertices $x, y \in V(G)$ if x and y belong to two different connected components of $G - X$.

Parameterized Complexity. An instance of a parameterized problem Π comprises an input I, which is an input of the classical instance of the problem, and an integer ℓ, which is called the parameter. A problem Π is said to be *fixed-parameter tractable* or in FPT if given an instance (I, ℓ) of Π, we can decide whether or not (I, ℓ) is a YES-instance of Π in time $f(\ell) \cdot |I|^{\mathcal{O}(1)}$, for some computable function f whose value depends only on ℓ.

A *kernelization* algorithm for Π is a polynomial-time algorithm that takes as input an instance (I, ℓ) of Π and returns an *equivalent* instance (I', ℓ') of Π with $|I'|, \ell' \leq f(\ell)$, where f is a function that depends only on the initial parameter ℓ. If such an algorithm exists for Π, we say that Π admits a kernel of *size* $f(\ell)$. If f is a polynomial or exponential function of ℓ, we say that Π admits a polynomial or exponential kernel, respectively. If Π is a graph problem, then I contains a graph, say G, and I' contains a graph, say G'. In this case, we say that Π admits a kernel with $f(\ell)$ vertices if the number of vertices of G' is at most $f(\ell)$.

It is typical to describe a kernelization algorithm as a series of reduction rules. A *reduction rule* is a polynomial time algorithm that takes as an input an instance of a problem and outputs another (usually reduced) instance. A reduction rule said to be *applicable* on an instance if the output instance is different from the input instance. A reduction rule is *safe* if the input instance is a YES-instance if and only if the output instance is a YES-instance. For more on parameterized complexity and related terminologies, we refer the reader to the recent book by Cygan et al. [23].

(Strong) Metric Dimension. A subset of vertices $S \subseteq V(G)$ *resolves* a pair of vertices $u, v \in V(G)$ if there exists a vertex $w \in S$ such that $d(w, u) \neq d(w, v)$. A subset of vertices $S \subseteq V(G)$ is a *resolving set* of G if it resolves all pairs of vertices $u, v \in V(G)$. A vertex $u \in V(G)$ is *distinguished* by a subset of vertices $S \subseteq V(G)$ if, for any $v \in V(G) \setminus \{u\}$, there exists a vertex $w \in S$ such that $d(w, u) \neq d(w, v)$. For an ordered subset of vertices

$S = \{s_1, \ldots, s_k\} \subseteq V(G)$ and a single vertex $u \in V(G)$, the *distance vector* of S with respect to u is $r(S|u) := (d(s_1, u), \ldots, d(s_k, u))$. The next observation is used throughout the paper.

▶ **Observation 1.** *Let G be a graph. For any (true or false) twins $u, v \in V(G)$ and any $w \in V(G) \setminus \{u, v\}$, $d(u, w) = d(v, w)$, and so, for any resolving set S of G, $S \cap \{u, v\} \neq \emptyset$.*

Proof. As $w \in V(G) \setminus \{u, v\}$, and u and v are (true or false) twins, the shortest (u, w)- and (v, w)-paths contain a vertex of $N := N(u) \setminus \{v\} = N(v) \setminus \{u\}$, and $d(u, w) = d(v, w)$. Hence, any resolving set S of G contains at least one of u and v. ◀

A vertex $s \in V(G)$ *strongly resolves* a pair of vertices $u, v \in V(G)$ if there exists a shortest path from u to s containing v, or a shortest path from v to s containing u. A subset $S \subseteq V(G)$ is a *strong resolving set* is every pair of vertices in $V(G)$ is strongly resolved by a vertex in S.

Geodetic Set. A subset $S \subseteq V(G)$ is a *geodetic set* if for every $u \in V(G)$, the following holds: there exist $s_1, s_2 \in S$ such that u lies on a shortest path from s_1 to s_2. The following simple observation is used throughout the paper. Recall that a vertex is *simplicial* if its neighborhood forms a clique.

▶ **Observation 2** ([20]). *If a graph G contains a simplicial vertex v, then v belongs to any geodetic set of G.*

Proof. Observe that v does not belong to any shortest path between any pair x, y of vertices (both distinct from v). ◀

This gives the following observation as an immediate corollary.

▶ **Observation 3.** *If a graph G contains a degree-1 vertex v, then v belongs to any geodetic set of G.*

3-Partitioned-3-SAT. Most of our lower bound proofs consist of reductions from the 3-Partitioned-3-SAT problem, a version of 3-SAT introduced in [74] and defined as follows.

> 3-Partitioned-3-SAT
> **Input:** A formula ψ in 3-CNF form, together with a partition of the set of its variables into three disjoint sets X^α, X^β, X^γ, with $|X^\alpha| = |X^\beta| = |X^\gamma| = n$, and such that no clause contains more than one variable from each of X^α, X^β, and X^γ.
> **Question:** Determine whether ψ is satisfiable.

The authors of [74] also proved the following.

▶ **Proposition 4** ([74, Theorem 3]). *Unless the ETH fails, 3-Partitioned-3-SAT does not admit an algorithm running in time $2^{o(n)}$.*

We will also use the following restricted version of the above problem.

> Exact-3-Partitioned-3-SAT
> **Input:** A formula ψ in 3-CNF form, together with a partition of the set of its variables into three disjoint sets X^α, X^β, X^γ, with $|X^\alpha| = |X^\beta| = |X^\gamma| = n$, and every clause contains exactly one variable from each of X^α, X^β, and X^γ.
> **Question:** Determine whether ψ is satisfiable.

For completeness, we repeat the polynomial-time reduction in [74] from 3-SAT to 3-Partitioned-3-SAT that increases the number of variables and clauses by a constant factor. Importantly, we make a simple change to adapt the proof for Exact-3-Partitioned-3-SAT.

▶ **Proposition 5.** *[74, Theorem 3] Unless the ETH fails, 3-Partitioned-3-SAT or Exact-3-Partitioned-3-SAT does not admit an algorithm running in time $2^{o(n)}$.*

Proof. Let ψ be a 3-SAT formula of m clauses and n variables. We can assume, without loss of generality, that every variable is used in some clause and every clause contains at least two literals. Suppose $X = \{x_1, \dots, x_n\}$ is the set of variables in ψ. We construct an equivalent instance ψ' of 3-Partitioned-3-SAT as follows:

- For every $i \in [n]$, we introduce three variables x_i^α, x_i^β, and x_i^γ, corresponding to the variable x_i, to ψ'.
- For every clause, e.g., $C = (x_i \vee \neg x_j \vee x_\ell)$, we introduce the clause $C' = (x_i^\alpha \vee \neg x_j^\beta \vee x_\ell^\gamma)$ to ψ'. In an analogous way, for every clause $C = (x_i \vee x_j)$, we introduce $C' = (x_i^\alpha \vee x_j^\beta)$.
- For every $i \in [n]$, we introduce the clauses $(\neg x_i^\alpha \vee x_i^\beta)$, $(\neg x_i^\beta \vee x_i^\gamma)$, and $(x_i^\alpha \vee \neg x_i^\gamma)$.

Define $X^\alpha = \{x_i^\alpha \mid i \in [n]\}$, and X^β, X^γ in the analogous way. Note that ψ' is a valid instance of 3-Partitioned-3-SAT as its variable set is divided into three equal parts, X^α, X^β, and X^γ, and each clause contains at most one variable from each of these parts. To see that ψ and ψ' are equivalent instances, consider a satisfying assignment $\pi : X \mapsto \{\texttt{True}, \texttt{False}\}$ for ψ. Consider the assignment $\pi' : X^\alpha \cup X^\beta \cup X^\gamma \mapsto \{\texttt{True}, \texttt{False}\}$ defined as follows: $\pi'(x_i^\alpha) = \pi'(x_i^\beta) = \pi'(x_i^\gamma) = \pi(x_i)$ for all $i \in [n]$. It is easy to verify that the assignment π' is a satisfying assignment for ψ'. In the reverse direction, consider a satisfying assignment $\pi' : X^\alpha \cup X^\beta \cup X^\gamma \mapsto \{\texttt{True}, \texttt{False}\}$ for ψ'. Note that the clauses added in the third step above are all satisfied if and only if the variables x_i^α, x_i^β, and x_i^γ share the same assignment, i.e., either all are `True` or all are `False`. Hence, $\pi'(x_i^\alpha) = \pi'(x_i^\beta) = \pi'(x_i^\gamma)$. It is easy to see that $\pi : X \mapsto \{\texttt{True}, \texttt{False}\}$, where $\pi(x_i) = \pi'(x_i^\alpha)$ for all $i \in [n]$, is a satisfying assignment for ψ. As the number of variables in ψ' is at most 3 times the number of variables in ψ, if 3-Partitioned-3-SAT admits an algorithm running in time $2^{o(n)}$, so does 3-SAT, which contradicts the ETH. This completes the first part of the proposition.

To prove the second part, we add the following steps to the above reduction.

- We add the variables x_0^α, x_0^β, x_0^γ to X^α, X^β, and X^γ, respectively.
- We add the following clauses:
  - $(\neg x_0^\alpha \vee \neg x_0^\beta \vee \neg x_0^\gamma)$,
  - $(x_0^\alpha \vee \neg x_0^\beta \vee \neg x_0^\gamma)$, $(\neg x_0^\alpha \vee x_0^\beta \vee \neg x_0^\gamma)$, $(\neg x_0^\alpha \vee \neg x_0^\beta \vee x_0^\gamma)$, and
  - $(\neg x_0^\alpha \vee x_0^\beta \vee x_0^\gamma)$, $(x_0^\alpha \vee \neg x_0^\beta \vee x_0^\gamma)$, $(x_0^\alpha \vee x_0^\beta \vee \neg x_0^\gamma)$.
  - For every clause that has only two literals, we add x_0^γ.

By the construction above, each clause that had only two literals contained literals corresponding to variables in X^α and X^β. Thus, ψ' is a valid instance of Exact-3-Partitioned-3-SAT. Now, it suffices to note that any satisfying assignment π' for ψ' must set x_0^α, x_0^β, and x_0^γ to `False`. Then, the other arguments are similar to those mentioned in the above paragraph. ◀

5 Metric Dimension: Lower Bound Regarding Diameter plus Treewidth

The aim of this section is to prove the following theorem.

▶ **Theorem 6.** *Unless the ETH fails, Metric Dimension does not admit an algorithm running in time $2^{f(\texttt{diam})^{o(\texttt{tw})}} \cdot n^{\mathcal{O}(1)}$ for any computable function $f : \mathbb{N} \mapsto \mathbb{N}$.*

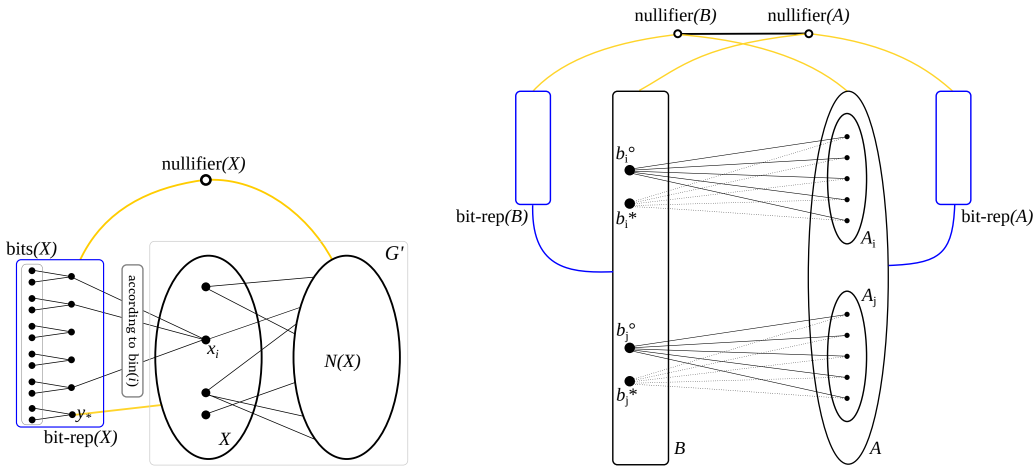

Figure 5 Set Identifying Gadget (left). The blue box represents bit-rep(X) and the yellow lines represent that every vertex in bit-rep(X) is adjacent to nullifier(X), nullifier(X) is adjacent to every vertex in $N(X)$, and $y_\star$ is adjacent to every vertex in X. Note that G' is not necessarily restricted to the graph induced by the vertices in $X \cup N(X)$. **Vertex Selector Gadget (right).** For $X \in \{B, A\}$, the blue box represents bit-rep(X), the blue link represents the connection with respect to the binary representation, and the yellow line represents that nullifier(X) is connected to every vertex in the set. The dotted lines highlight the absence of edges.

To this end, we present a reduction from 3-Partitioned-3-SAT to Metric Dimension. The reduction takes as input an instance ψ of 3-Partitioned-3-SAT on $3n$ variables (see Section 4 for a definition of this problem), and returns (G, k) as an instance of Metric Dimension such that $\mathtt{tw}(G) = \mathcal{O}(\log(n))$ and $\mathtt{diam}(G) = \mathcal{O}(1)$. Before presenting the reduction, we first introduce some preliminary tools.

5.1 Preliminary Tools

5.1.1 Set Identifying Gadget

Suppose that we are given a graph G' and a subset $X \subseteq V(G')$ of its vertices. Further, suppose that we want to add a vertex set X^+ to G' in order to obtain a new graph G with the following properties. We want that each vertex in $X \cup X^+$ will be distinguished by vertices in X^+ that must be in any resolving set S of G, and no vertex in X^+ can resolve any "critical pair" of vertices in $V(G)$ (critical pairs will be defined in the next subsection).

We refer to the graph induced by the vertices of X^+, along with the edges connecting X^+ to G', as the Set Identifying Gadget for the set X.

Given a graph G' and a non-empty subset $X \subseteq V(G')$ of its vertices, to construct such a graph G, we add vertices and edges to G' as follows:

- The vertex set X^+ that we are aiming to add is the union of a set bit-rep(X) and a special vertex denoted by nullifier(X).
- First, let $X = \{x_i \mid i \in [|X|]\}$, and set $q := \lceil \log(|X| + 2) \rceil + 1$. We select this value for q to *(1)* uniquely represent each integer in $[|X|]$ by its bit-representation in binary (note that we start from 1 and not 0), *(2)* ensure that the only vertex whose bit-representation contains all 1's is nullifier(X), and *(3)* reserve one spot for an additional vertex $y_\star$.
- For every $i \in [q]$, add three vertices y_i^a, y_i, y_i^b, and add the path (y_i^a, y_i, y_i^b).
- Add three vertices $y_\star^a, y_\star, y_\star^b$, and add the path $(y_\star^a, y_\star, y_\star^b)$. Add all the edges to make

$\{y_i \mid i \in [q]\} \cup \{y_\star\}$ into a clique. Make $y_\star$ adjacent to each vertex $v \in X$. We denote $\mathsf{bit\text{-}rep}(X) = \{y_i, y_i^a, y_i^b \mid i \in [q]\} \cup \{y_\star, y_\star^a, y_\star^b\}$ and its subset $\mathsf{bits}(X) = \{y_i^a, y_i^b \mid i \in [q]\} \cup \{y_\star^a, y_\star^b\}$ for convenience in a later case analysis.

- For every integer $j \in [|X|]$, let $\mathsf{bin}(j)$ denote the binary representation of j using q bits. Connect x_j with y_i if the i^{th} bit (going from left to right) in $\mathsf{bin}(j)$ is 1.
- Add a vertex, denoted by $\mathsf{nullifier}(X)$, and make it adjacent to every vertex in $\{y_i \mid i \in [q]\} \cup \{y_\star\}$. One can think of the vertex $\mathsf{nullifier}(X)$ as the only vertex whose bit-representation contains all 1's.
- For every vertex $u \in V(G) \setminus (X \cup X^+)$ such that u is adjacent to some vertex in X, add an edge between u and $\mathsf{nullifier}(X)$. We add this vertex to ensure that vertices in $\mathsf{bit\text{-}rep}(X)$ do not resolve critical pairs in $V(G)$.

This completes the construction of G. The properties of G are not proven yet, but just given as an intuition behind its construction. See Figure 5 for an illustration.

5.1.2 Gadget to Add Critical Pairs

Any resolving set needs to resolve *all* pairs of vertices in the input graph. As we will see, some pairs, which we call critical pairs, are harder to resolve than others. In fact, the non-trivial part will be to resolve all of the critical pairs.

Suppose that we need to have many critical pairs in a graph G, say $\langle c_i^\circ, c_i^\star\rangle$ for every $i \in [m]$ for some $m \in \mathbb{N}$. Define $C := \{c_i^\circ, c_i^\star \mid i \in [m]\}$. We then add $\mathsf{bit\text{-}rep}(C)$ and $\mathsf{nullifier}(C)$ as mentioned above (taking C as the set X), but the connection across $\{c_i^\circ, c_i^\star\}$ and $\mathsf{bit\text{-}rep}(C)$ is defined by $\mathsf{bin}(i)$, i.e., connect both c_i° and $c_i^\star$ with the j-th vertex of $\mathsf{bit\text{-}rep}(C)$ if the j^{th} digit (going from left to right) in $\mathsf{bin}(i)$ is 1. Hence, $\mathsf{bit\text{-}rep}(C)$ can resolve any pair of the form $\langle c_i^\circ, c_\ell^\star\rangle$, $\langle c_i^\circ, c_\ell^\circ\rangle$, or $\langle c_i^\star, c_\ell^\star\rangle$ as long as $i \neq \ell$. As before, $\mathsf{bit\text{-}rep}(C)$ can also resolve all pairs with one vertex in $C \cup \mathsf{bit\text{-}rep}(C) \cup \{\mathsf{nullifier}(C)\}$, but no critical pair of vertices. Again, when these facts will be used, they will be proven formally.

5.1.3 Vertex Selector Gadgets

Suppose that we are given a collection of sets $A_1, A_2, \dots, A_q$ of vertices in a graph G, and we want to ensure that any resolving set of G includes at least one vertex from A_i for every $i \in [q]$. In the following, we construct a gadget that achieves a slightly weaker objective.

- Let $A = \bigcup_{i\in[q]} A_i$. Add a set identifying gadget for A as mentioned in Subsection 5.1.1.
- For every $i \in [q]$, add two vertices b_i° and $b_i^\star$. Use the gadget mentioned in Subsection 5.1.2 to make all the pairs of the form $\langle b_i^\circ, b_i^\star\rangle$ critical pairs.
- For every $a \in A_i$, add an edge (a, b_i°). We highlight that we do not make a adjacent to $b_i^\star$ by a dotted line in Figure 5. Also, add the edges $(a, \mathsf{nullifier}(B))$, $(b_i^\circ, \mathsf{nullifier}(A))$, $(b_i^\star, \mathsf{nullifier}(A))$, and $(\mathsf{nullifier}(A), \mathsf{nullifier}(B))$.

This completes the construction.

Note that the only vertices that can resolve a critical pair $\langle b_i^\circ, b_i^\star\rangle$, apart from b_i° and $b_i^\star$, are the vertices in A_i. Hence, every resolving set contains at least one vertex in $\{b_i^\circ, b_i^\star\} \cup A_i$. Again, when used, these facts will be proven formally.

5.1.4 Set Representation

For a positive integer p, define $\mathcal{F}_p$ as the collection of subsets of $[2p]$ that contains exactly p integers. We critically use the fact that no set in $\mathcal{F}_p$ is contained in any other set in $\mathcal{F}_p$ (such a collection of sets are called a *Sperner family*). Let ℓ be a positive integer such that

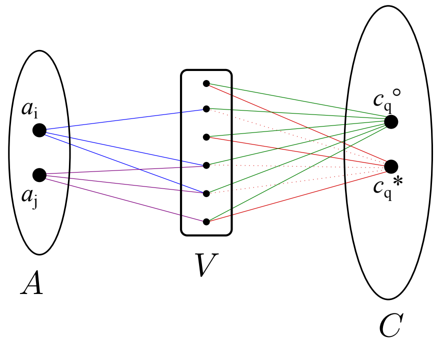

Figure 6 A toy example to illustrate the application of set-rep. See Subsection 5.1. Suppose that $\phi(q) = i$, i.e., we want to add gadgets such that only a_i in A can resolve the critical pair $\langle c_q^\circ, c_q^\star \rangle$. Suppose that the vertices in V are indexed from top to bottom and $\mathsf{set\text{-}rep}(i) = \{2, 4, 5\}$. By the construction, the only vertices in V that $c_q^\star$ is *not* adjacent to are v_2, v_4, and v_5 (this fact is highlighted with red-dotted edges). Thus, $\text{dist}(a_i, c_q^\circ) = 2$ and $\text{dist}(a_i, c_q^\star) > 2$, and hence, a_i resolves the critical pair $\langle c_q^\circ, c_q^\star \rangle$. For any other vertex in A, say a_j, $\mathsf{set\text{-}rep}(j) \setminus \mathsf{set\text{-}rep}(i)$ is a non-empty set. Hence, there are shortest paths from a_j to c_q°, and a_j to $c_q^\star$ through the vertices in V with indices in $\mathsf{set\text{-}rep}(j) \setminus \mathsf{set\text{-}rep}(i)$. This implies that $\text{dist}(a_j, c_q^\circ) = \text{dist}(a_j, c_q^\star) = 2$ and a_j cannot resolve the pair $\langle c_q^\circ, c_q^\star \rangle$. The sets $\mathsf{bit\text{-}rep}(X)$ and $\mathsf{nullifier}(X)$ are omitted for $X \in \{A, V, C\}$.

$\ell \leq \binom{2p}{p}$. We define $\mathsf{set\text{-}rep} : [\ell] \mapsto \mathcal{F}_p$ as a one-to-one function by arbitrarily assigning a set in $\mathcal{F}_p$ to an integer in $[\ell]$. By the asymptotic estimation of the central binomial coefficient, $\binom{2p}{p} \sim \frac{4^p}{\sqrt{\pi \cdot p}}$ [59]. To get the upper bound of p, we scale down the asymptotic function and have $\ell \leq \frac{4^p}{2^p} = 2^p$. Thus, $p = \mathcal{O}(\log \ell)$.

We mention an application of such a function in the context of Metric Dimension. Suppose that we have a "large" collection of vertices, say $A = \{a_1, a_2, \ldots, a_\ell\}$, and a "large" collection of critical pairs $C = \{\langle c_1^\circ, c_1^\star \rangle, \langle c_2^\circ, c_2^\star \rangle, \ldots, \langle c_m^\circ, c_m^\star \rangle\}$. Moreover, we are given an injective function $\phi : [m] \mapsto [\ell]$. The objective is to design a gadget such that only $a_{\phi(q)} \in A$ can resolve a critical pair $\langle c_q^\circ, c_q^\star \rangle \in C$ for any $q \in [m]$, while keeping the treewidth of this part of the graph of order $\mathcal{O}(\log(|A|))$. We add the following vertices and edges in order to achieve this objective.

- Add vertices and edges as mentioned in Subsection 5.1.1 and in Subsection 5.1.2, respectively, to identify the set A and to add critical pairs in C.
- Add a *validation portal*, a clique on $2p$ vertices, denoted by $V = \{v_1, v_2, \ldots, v_{2p}\}$, and vertices and edges to identify it.
- For every $i \in [\ell]$ and for every $p' \in \mathsf{set\text{-}rep}(i)$, add the edge $(a_i, v_{p'})$.
- For every critical pair $\langle c_q^\circ, c_q^\star \rangle$, make c_q° adjacent to every vertex in V, and add every edge of the form $(c_q^\star, v_{p'})$ for $p' \in [2p] \setminus \mathsf{set\text{-}rep}(\phi(q))$. Note that the vertices in V that are indexed using integers in $\mathsf{set\text{-}rep}(\phi(q))$ are *not* adjacent with $c_q^\star$.

See Figure 6 for an illustration.

5.2 Reduction

Consider an instance ψ of 3-Partitioned-3-SAT, with $X^\alpha, X^\beta, X^\gamma$ the partition of the variable set. From ψ, we construct the graph G as follows. We describe the construction of X^α, with the constructions for X^β and X^γ being analogous. We rename the variables in X^α to x_i^α for $i \in [n]$.

- For every variable x_i^{α}, we add a pair $\langle x_i^{\alpha,\circ}, x_i^{\alpha,\star}\rangle$ of vertices. We add vertices and edges as mentioned in Subsection 5.1.2 to make all pairs of the form $\langle x_i^{\alpha,\circ}, x_i^{\alpha,\star}\rangle$ critical in the graph G. We denote $X^{\alpha} = \{x_i^{\alpha,\circ}, x_i^{\alpha,\star} \mid i \in [n]\}$ as the collection of vertices in the critical pairs. We remark that we do *not* convert X^{α} into a clique.
- For every variable x_i^{α}, we add the vertices t_{2i}^{α} and f_{2i-1}^{α}. Formally, $A^{\alpha} = \{t_{2i}^{\alpha}, f_{2i-1}^{\alpha} \mid i \in [n]\}$, and hence, $|A^{\alpha}| = 2n$. We add vertices and edges as mentioned in Subsection 5.1.1 in order to identify the set A^{α} in G.
- We would like that any resolving set contains at least one vertex in $\{t_{2i}^{\alpha}, f_{2i-1}^{\alpha}\}$ for every $i \in [n]$, but instead we add the construction mentioned in Subsection 5.1.3 that achieves the slightly weaker objective as mentioned there. As before, instead of adding two new vertices, we use $\langle x_i^{\alpha,\circ}, x_i^{\alpha,\star}\rangle$ as the necessary critical pair. Formally, for every $i \in [n]$, we add the edges $(x_i^{\alpha,\circ}, t_{2i}^{\alpha})$ and $(x_i^{\alpha,\circ}, f_{2i-1}^{\alpha})$. We add edges to make $\mathsf{nullifier}(X^{\alpha})$ adjacent to every vertex in A^{α}, and $\mathsf{nullifier}(A^{\alpha})$ adjacent to every vertex in X^{α}. Also, we add the edge $(\mathsf{nullifier}(X^{\alpha}), \mathsf{nullifier}(A^{\alpha}))$.
- Let p be the smallest positive integer such that $2n \leq \binom{2p}{p}$. In particular, $p = \mathcal{O}(\log n)$. Moreover, define $\mathsf{set\text{-}rep} : [2n] \mapsto \mathcal{F}_p$ as mentioned in Subsection 5.1.
- We add a *validation portal*, a clique on $2p$ vertices, denoted by $V^{\alpha} = \{v_1^{\alpha}, v_2^{\alpha}, \ldots, v_{2p}^{\alpha}\}$. We add vertices and edges to identify V^{α} as mentioned in Subsection 5.1.1. We add the edge $(\mathsf{nullifier}(V^{\alpha}), \mathsf{nullifier}(A^{\alpha}))$ and make $\mathsf{nullifier}(A^{\alpha})$ adjacent to every vertex in V^{α}. We note that we *do not* add edges across $\mathsf{nullifier}(V^{\alpha})$ and A^{α}.
- We add edges across A^{α} and the validation portal as follows: for each $i \in [n]$, we add the edge $(t_{2i}^{\alpha}, v_{p'}^{\alpha})$ for every $p' \in \mathsf{set\text{-}rep}(2i)$. Similarly, for each $i \in [n]$, we add the edge $(f_{2i-1}^{\alpha}, v_{p'}^{\alpha})$ for every $p' \in \mathsf{set\text{-}rep}(2i-1)$.
We repeat the above steps to construct $X^{\beta}, A^{\beta}, V^{\beta}$, $X^{\gamma}, A^{\gamma}, V^{\gamma}$, and their related vertices and edges.
- For every clause C_q in ψ, we introduce a pair $\langle c_q^{\circ}, c_q^{\star}\rangle$ of vertices. We add vertices and edges to make each pair of the form $\langle c_q^{\circ}, c_q^{\star}\rangle$ a critical pair as mentioned in Subsection 5.1.2. Let C be the collection of the vertices in such pairs.
- We add edges across C and the portals as follows. Consider a clause C_q in ψ and the corresponding critical pair $\langle c_q^{\circ}, c_q^{\star}\rangle$ in C. Suppose $\delta \in \{\alpha, \beta, \gamma\}$. As ψ is an instance of 3-Partitioned-3-SAT, there is at most one variable in X^{δ} that appears in C_q. Suppose that variable is x_i^{δ} for some $i \in [n]$.
We add all edges of the form $(v_{p'}^{\delta}, c_q^{\circ})$ for every $p' \in [2p]$. If x_i^{δ} appears as a positive literal in C_q, then we add the edge $(v_{p'}^{\delta}, c_q^{\star})$ for every $p' \in [2p] \setminus \mathsf{set\text{-}rep}(2i)$ (which corresponds to t_{2i}^{δ}). If x_i^{δ} appears as a negative literal in C_q, then we add the edge $(v_{p'}^{\delta}, c_q^{\star})$ for every $p' \in [2p] \setminus \mathsf{set\text{-}rep}(2i-1)$ (which corresponds to f_{2i-1}^{δ}). We remark that if x_i^{δ} appears as a positive (negative, respectively) literal in C_q, then the vertices in V^{δ} whose indices are in $\mathsf{set\text{-}rep}(2i)$ ($\mathsf{set\text{-}rep}(2i-1)$, respectively) are *not adjacent* to $c_q^{\star}$. If there is no variable in X^{δ} that appears in C_q, then we make every vertex in V^{δ} adjacent to both c_q° and $c_q^{\star}$. Finally, we add the edge $(\mathsf{nullifier}(V^{\delta}), \mathsf{nullifier}(C))$. See Figure 7.

This concludes the construction of G. The reduction returns (G, k) as an instance of Metric Dimension where

$$k = 3\cdot(n+(\lceil\log(|X^{\alpha}|/2+2)\rceil+1)+(\lceil\log(|A^{\alpha}|+2)\rceil+1)+(\lceil\log(|V^{\alpha}|+2)\rceil+1))+\lceil\log(|C|+2)\rceil+1.$$

5.3 Correctness of the Reduction

Suppose, given an instance ψ of 3-Partitioned-3-SAT, that the reduction returns (G, k) as an instance of Metric Dimension. We first prove the following lemma, which will be

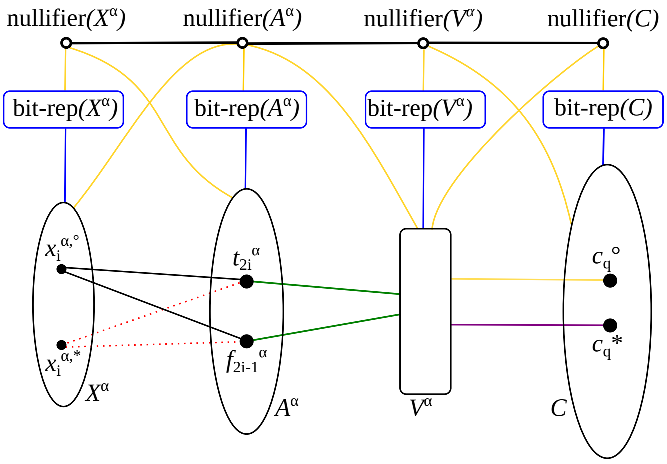

Figure 7 Overview of the reduction. For any set $X \in \{X^\alpha, A^\alpha, V^\alpha, C\}$, the blue rectangle attached to it via the blue edge represents $\mathsf{bit\text{-}rep}(X)$. The yellow thick line represents that vertex is connected to every vertex in the set the edge goes to. Note that $\mathsf{nullifier}(V^\alpha)$ is *not* adjacent to any vertex in A^α. Green edges denote adjacencies with respect to $\mathsf{set\text{-}rep}$, i.e., t^α_{2i} is adjacent to $v_j \in V^\alpha$ if $j \in \mathsf{set\text{-}rep}(2i)$. The same holds for f_{2i-1} for all $i \in [n]$. Purple lines also indicate adjacencies with respect to $\mathsf{set\text{-}rep}$, but in a complementary way, i.e., if $x_i \in c_q$, then, for all $p' \in [2p] \setminus \mathsf{set\text{-}rep}(2i)$, we have that $(v^\alpha_{p'}, c^\star_q) \in E(G)$, and if $\overline{x}_i \in c_q$, then, for all $p' \in [2p] \setminus \mathsf{set\text{-}rep}(2i-1)$, we have that $(v^\alpha_{p'}, c^\star_q) \in E(G)$.

helpful in proving the correctness of the reduction.

▶ **Lemma 7.** *For any resolving set S of G and for all $X \in \{X^\delta, A^\delta, V^\delta, C\}$ and $\delta \in \{\alpha, \beta, \gamma\}$,*

1. *S contains at least one vertex from each pair of false twins in $\mathsf{bits}(X)$.*
2. *Vertices in $\mathsf{bits}(X) \cap S$ resolve any non-critical pair of vertices $\langle u, v\rangle$ when $u \in X \cup X^+$ and $v \in V(G)$.*
3. *Vertices in $X^+ \cap S$* cannot *resolve any critical pair of vertices $\langle x_i^{\delta',\circ}, x_i^{\delta',\star}\rangle$ nor $\langle c^\circ_q, c^\star_q\rangle$ for all $i \in [n]$, $\delta' \in \{\alpha, \beta, \gamma\}$, and $q \in [m]$.*

Proof. 1. By Observation 1, the statement follows for all $X \in \{X^\delta, A^\delta, V^\delta, C\}$ and $\delta \in \{\alpha, \beta, \gamma\}$.

2. For all $X \in \{X^\delta, A^\delta, V^\delta, C\}$ and $\delta \in \{\alpha, \beta, \gamma\}$, note that $\mathsf{nullifier}(X)$ is distinguished by $S \cap \mathsf{bits}(X)$ since it is the only vertex in G that is at distance 2 from every vertex in $\mathsf{bits}(X)$. We now do a case analysis for the remaining non-critical pairs of vertices $\langle u, v\rangle$ assuming that $\mathsf{nullifier}(X) \notin \{u, v\}$ (also, suppose that both u and v are not in S, as otherwise, they are obviously distinguished):

   Case i: $u, v \in X \cup X^+$.

   Case i(a): $u, v \in X$ or $u, v \in \mathsf{bit\text{-}rep}(X) \setminus \mathsf{bits}(X)$. In the first case, let j be the digit in the binary representation of the subscript of u that is not equal to the j^{th} digit in the binary representation of the subscript of v (such a j exists since $\langle u, v\rangle$ is not a critical pair). In the second case, without loss of generality, let $u = y_i$ and $v = y_j$. By the first item of the statement of the lemma (1.), without loss of generality, $y^a_j \in S \cap \mathsf{bits}(X)$. Then, in both cases, $d(y^a_j, u) \neq d(y^a_j, v)$.

   Case i(b): $u \in X$ and $v \in \mathsf{bit\text{-}rep}(X)$. Without loss of generality, $y^a_\star \in S \cap \mathsf{bits}(X)$ (by 1.). Then, $d(y^a_\star, u) = 2$ and, for all $v \in \mathsf{bits}(X) \setminus \{y^b_\star\}$, $d(y^a_\star, v) = 3$. Without loss of generality, let y_i be adjacent to u and let $y^a_i \in S \cap \mathsf{bits}(X)$ (by 1.). Then, for

$v = y^b_\star$, $3 = d(y^a_i, v) \neq d(y^a_i, u) = 2$. If $v \in \mathsf{bit\text{-}rep}(X) \setminus \mathsf{bits}(X)$, then, without loss of generality, $v = y_j$ and $y^a_j \in S \cap \mathsf{bits}(X)$ (by 1.), and $1 = d(y^a_j, v) < d(y^a_j, u)$.

Case i(c): $u, v \in \mathsf{bits}(X)$. Without loss of generality, $u = y^b_i$ and $y^a_i \in S$ (by 1.). Then, $2 = d(y^a_i, u) \neq d(y^a_i, v) = 3$.

Case i(d): $u \in \mathsf{bits}(X)$ and $v \in \mathsf{bit\text{-}rep}(X) \setminus \mathsf{bits}(X)$. Without loss of generality, $v = y_i$ and $y^a_i \in S$ (by 1.). Then, $1 = d(y^a_i, v) < d(y^a_i, u)$.

Case ii: $u \in X \cup X^+$ and $v \in V(G) \setminus (X \cup X^+)$. For each $u \in X \cup X^+$, there exists $w \in \mathsf{bits}(X) \cap S$ such that $d(u, w) \leq 2$, while, for each $v \in V(G) \setminus (X \cup X^+)$ and $w \in \mathsf{bits}(X) \cap S$, we have $d(v, w) \geq 3$.

3. For all $X \in \{X^\delta, A^\delta, V^\delta\}$, $\delta \in \{\alpha, \beta, \gamma\}$, $u \in X^+$, $v \in \{c^\circ_q, c^\star_q\}$, and $q \in [m]$, we have that $d(u, v) = d(u, \mathsf{nullifier}(V^\delta)) + 1$. Further, for $X = C$ and all $u \in X^+$ and $q \in [m]$, either $d(u, c^\circ_q) = d(u, c^\star_q) = 1$, $d(u, c^\circ_q) = d(u, c^\star_q) = 2$, or $d(u, c^\circ_q) = d(u, c^\star_q) = 3$ by the construction in Subsection 5.1.2 and since $\mathsf{bit\text{-}rep}(X) \setminus \mathsf{bits}(X)$ is a clique. Hence, for all $X \in \{X^\delta, A^\delta, V^\delta, C\}$ and $\delta \in \{\alpha, \beta, \gamma\}$, vertices in $X^+ \cap S$ cannot resolve a pair of vertices $\langle c^\circ_q, c^\star_q \rangle$ for any $q \in [m]$.

   For all $\delta \in \{\alpha, \beta, \gamma\}$, if $v \in X^\delta$, then, for all $X \in \{X^{\delta'}, A^{\delta'}, V^{\delta'}, C\}$, $\delta' \in \{\alpha, \beta, \gamma\}$ such that $\delta \neq \delta'$, and $u \in X^+$, we have that $d(u, v) = d(u, \mathsf{nullifier}(A^\delta)) + 1$. Similarly, for all $\delta \in \{\alpha, \beta, \gamma\}$, if $v \in X^\delta$, then, for all $X \in \{A^\delta, V^\delta\}$ and $u \in X^+$, we have that $d(u, v) = d(u, \mathsf{nullifier}(A^\delta)) + 1$. Lastly, for each $\langle x^{\delta,\circ}_i, x^{\delta,\star}_i \rangle$, $\delta \in \{\alpha, \beta, \gamma\}$, and $i \in [n]$, if $X = X^\delta$, then, for all $u \in X^+$, either $d(u, x^{\delta,\circ}_i) = d(u, x^{\delta,\star}_i) = 1$, $d(u, x^{\delta,\circ}_i) = d(u, x^{\delta,\star}_i) = 2$, or $d(u, x^{\delta,\circ}_i) = d(u, x^{\delta,\star}_i) = 3$ by the construction in Subsection 5.1.2 and since $\mathsf{bit\text{-}rep}(X) \setminus \mathsf{bits}(X)$ is a clique. ◀

▶ **Lemma 8.** *If ψ is a satisfiable 3-Partitioned-3-SAT formula, then G admits a resolving set of size k.*

Proof. Suppose $\pi : X^\alpha \cup X^\beta \cup X^\gamma \mapsto \{\texttt{True}, \texttt{False}\}$ is a satisfying assignment for ψ. We construct a resolving set S of size k for G using this assignment.

For every $\delta \in \{\alpha, \beta, \gamma\}$ and $i \in [n]$, if $\pi(x^\delta_i) = \texttt{True}$, then let $t^\delta_{2i} \in S$, and otherwise, let $f^\delta_{2i-1} \in S$. For every $X \in \{B^\delta, A^\delta, V^\delta, C\}$ and $\delta \in \{\alpha, \beta, \gamma\}$, add one vertex from each pair of false twins in $\mathsf{bits}(X)$ to S. Note that the size of S is k.

In the remaining part of the proof, we show that S is a resolving set of G. First, we prove that all critical pairs are resolved by S in the following claim.

▷ Claim 9. All critical pairs are resolved by S.

Proof. For each $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$, the critical pair $\langle x^{\delta,\circ}_i, x^{\delta,\star}_i \rangle$ is resolved by the vertex $S \cap A^\delta$ by the construction. For each $q \in [m]$, the clause C_q is satisfied by the assignment π. Thus, there is a variable, say x_i in C_q, that satisfies C_q according to π. If x_i appears positively in C_q, then $t^\delta_{2i} \in S$ resolves the critical pair $\langle c^\circ_q, c^\star_q \rangle$ since $d(t^\alpha_{2i}, c^\circ_q) = 2 < d(t^\alpha_{2i}, c^\star_q)$ by the construction. Similarly, if x_i appears negatively in C_q, then $f^\delta_{2i-1} \in S$ resolves the critical pair $\langle c^\circ_q, c^\star_q \rangle$ since $d(f^\alpha_{2i-1}, c^\circ_q) = 2 < d(f^\alpha_{2i-1}, c^\star_q)$ by the construction. Thus, every critical pair $\langle c^\circ_q, c^\star_q \rangle$ is resolved by S. ◁

Then, every vertex pair in $V(G)$ is resolved by S by Claim 9 in conjunction with the second item of the statement of Lemma 7. ◀

▶ **Lemma 10.** *If G admits a resolving set of size k, then ψ is a satisfiable 3-Partitioned-3-SAT formula.*

Proof. Assume that G admits a resolving set S of size k. First, we prove some properties regarding S. By the first item of the statement of Lemma 7, for each $\delta \in \{\alpha, \beta, \gamma\}$, we have

$$|S \cap \mathsf{bits}(X^\delta)| \geq \lceil \log(|X^\delta|/2 + 2) \rceil + 1, \quad |S \cap \mathsf{bits}(V^\delta)| \geq \lceil \log(|V^\delta| + 2) \rceil + 1,$$
$$|S \cap \mathsf{bits}(A^\delta)| \geq \lceil \log(|A^\delta| + 2) \rceil + 1, \quad |S \cap \mathsf{bits}(C)| \geq \lceil \log(|C| + 2) \rceil + 1.$$

Hence, any resolving set S of G already has size at least

$$3 \cdot ((\lceil \log(|X^\alpha|/2+2) \rceil + 1) + (\lceil \log(|A^\alpha|+2) \rceil + 1) + (\lceil \log(|V^\alpha|+2) \rceil + 1)) + \lceil \log(|C|+2) \rceil + 1.$$

Now, for each $\delta \in \{\alpha, \beta, \gamma\}$ and $i \in [n]$, consider the critical pair $\langle x_i^{\delta,\circ}, x_i^{\delta,\star} \rangle$. By the construction mentioned in Subsection 5.1.2, only $v \in \{t_{2i}^\delta, f_{2i-1}^\delta, x_i^{\delta,\circ}, x_i^{\delta,\star}\}$ resolves a pair $\langle x_i^{\delta,\circ}, x_i^{\delta,\star} \rangle$. Indeed, for all $X \in \{X^{\delta'}, A^{\delta'}, V^{\delta'}, C\}$ and $\delta' \in \{\alpha, \beta, \gamma\}$, no vertex in X^+ can resolve such a pair by the third item of the statement of Lemma 7. Also, for all $X \in \{A^{\delta''}, A^\delta \setminus \{t_{2i}^\delta, f_{2i-1}^\delta\}, V^{\delta'}, C\}$, $\delta' \in \{\alpha, \beta, \gamma\}$, $\delta'' \in \{\alpha, \beta, \gamma\}$ such that $\delta \neq \delta''$, and $u \in X$, we have that $d(u, x_i^{\delta,\circ}) = d(u, x_i^{\delta,\star}) = d(u, \mathsf{nullifier}(A^\delta)) + 1$. Hence, since any resolving set S of G of size at most k can only admit at most another $3n$ vertices, we get that equality must in fact hold in every one of the aforementioned inequalities, and any resolving set S of G of size at most k contains one vertex from $\{t_{2i}^\delta, f_{2i-1}^\delta, x_i^{\delta,\circ}, x_i^{\delta,\star}\}$ for all $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$. Hence, any resolving set S of G of size at most k is actually of size exactly k.

Next, we construct an assignment $\pi : X^\alpha \cup X^\beta \cup X^\gamma \to \{\texttt{True}, \texttt{False}\}$ in the following way. For each $\delta \in \{\alpha, \beta, \gamma\}$ and $i \in [n]$, if $t_{2i}^\delta \in S$, then set $\pi(x_i^\delta) := \texttt{True}$, and if $f_{2i-1}^\delta \in S$, then set $\pi(x_i^\delta) := \texttt{False}$. For any $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$, if $S \cap \{t_{2i}^\delta, f_{2i-1}^\delta\} = \emptyset$, then one of $x_i^{\delta,\circ}, x_i^{\delta,\star}$ is in S, and we can use an arbitrary assignment of the variable x_i^δ.

We contend that the constructed assignment π satisfies every clause in C. Since S is a resolving set, it follows that, for every clause $c_q \in C$, there exists $v \in S$ such that $d(v, c_q^\circ) \neq d(v, c_q^\star)$. Note that, for any v in $\mathsf{bits}(A^\delta), \mathsf{bits}(X^\delta), \mathsf{bits}(V^\delta)$ for any $\delta \in \{\alpha, \beta, \gamma\}$ or in $\mathsf{bits}(C)$, we have $d(v, c_i^\circ) = d(v, c_i^\star)$ by the third item of the statement of Lemma 7. Further, for any $v \in X^\delta$ and any $\delta \in \{\alpha, \beta, \gamma\}$, we have that $d(v, c_q^\circ) = d(v, c_q^\star) = d(v, \mathsf{nullifier}(V^\delta)) + 1$. Thus, $v \in S \cap \bigcup_{\delta \in \{\alpha,\beta,\gamma\}} A^\delta$. Without loss of generality, suppose that c_q° and $c_q^\star$ are resolved by t_{2i}^α. So, $d(t_{2i}^\alpha, c_i^\circ) \neq d(t_{2i}^\alpha, c_i^\star)$. By the construction, the only case where $d(t_{2i}^\alpha, c_i^\circ) \neq d(t_{2i}^\alpha, c_i^\star)$ is when C_q contains a variable $x_i \in X^\alpha$ and $\pi(x_i)$ satisfies C_q. Thus, we get that the clause C_q is satisfied by the assignment π.

Since S resolves all pairs $\langle c_q^\circ, c_q^\star \rangle$ in $V(G)$, then the assignment π constructed above indeed satisfies every clause c_q, completing the proof. ◀

Proof of Theorem 6. In Subsection 5.2, we presented a reduction that takes an instance ψ of 3-Partitioned-3-SAT and returns an equivalent instance (G, k) of Metric Dimension (by Lemmas 8 and 10) in polynomial time. Now, consider the set

$$Z = \{V^\delta \cup X^+ \mid X \in \{X^\delta, A^\delta, V^\delta, C\}, \delta \in \{\alpha, \beta, \gamma\}\}.$$

It is easy to verify that $|Z| = \mathcal{O}(\log(n))$ and $G - Z$ is a collection of P_3's and isolated vertices. Hence, $\texttt{tw}(G)$, $\texttt{fvs}(G)$, and $\texttt{td}(G)$ are upper bounded by $\mathcal{O}(\log(n))$. It is also easy to see that the diameter of the graph is bounded by a constant. Hence, if there is an algorithm for Metric Dimension that runs in time $2^{f(\texttt{diam})^{o(\texttt{tw})}}$ (or $2^{f(\texttt{diam})^{o(\texttt{fvs})}}$ or $2^{f(\texttt{diam})^{o(td)}}$), then there is an algorithm solving 3-Partitioned-3-SAT running in time $2^{o(n)}$, which by Proposition 5 contradicts the ETH. ◀

6 Geodetic Set: Lower Bound Regarding Diameter plus Treewidth

The aim of this section is to prove the following theorem.

▶ **Theorem 11.** *Unless the ETH fails, Geodetic Set does not admit an algorithm running in time $2^{f(\mathtt{diam})^{o(\mathtt{tw})}} \cdot n^{\mathcal{O}(1)}$ for any computable function $f : \mathbb{N} \mapsto \mathbb{N}$.*

As in the previous section, we present a different reduction from 3-Partitioned-3-SAT (see Section 4) to Geodetic Set. The reduction takes as input an instance ψ of 3-Partitioned-3-SAT on $3n$ variables and returns (G, k) as an instance of Geodetic Set such that $\mathtt{tw}(G) = \mathcal{O}(\log(n))$ and $\mathtt{diam}(G) = \mathcal{O}(1)$. We rely on the tool of set representation introduced in Section 5.1.4. For convenience, we recall it in the next subsection and describe how we apply it in the reduction to prove Theorem 11.

6.1 Preliminary Tool: Set Representation

For a positive integer p, define $\mathcal{F}_p$ as the collection of subsets of $[2p]$ that contains exactly p integers. We critically use the fact that no set in $\mathcal{F}_p$ is contained in any other set in $\mathcal{F}_p$ (such a collection of sets is called a *Sperner family*). Let ℓ be a positive integer such that $\ell \leq \binom{2p}{p}$. We define set-rep : $[\ell] \mapsto \mathcal{F}_p$ as a one-to-one function by arbitrarily assigning a set in $\mathcal{F}_p$ to an integer in $[\ell]$. By the asymptotic estimation of the central binomial coefficient, $\binom{2p}{p} \sim \frac{4^p}{\sqrt{\pi \cdot p}}$ [59]. To get the upper bound of p, we scale down the asymptotic function and have $\ell \leq \frac{4^p}{2^p} = 2^p$. Thus, $p = \mathcal{O}(\log \ell)$.

We will apply the existence of such a function in the context of Geodetic Set. Suppose we have a "large" collection of vertices, say $A = \{a_1, a_2, \ldots, a_\ell\}$, and a "large" collection of vertices $C = \{c_1, c_2, \ldots, c_m\}$. Moreover, we are given a function $\phi : [m] \mapsto [\ell]$. The basic idea is to design gadgets such that c_q is only covered by the shortest path from $a_{\phi(q)} \in A$ to c_q^b (c_q^b is forced to be chosen in the geodetic set) for any $q \in [m]$, while keeping the treewidth of this part of the graph of order $\mathcal{O}(\log(|A|))$. To do so, we create a "small" intermediate set V (of size $\mathcal{O}(\log(|A|))$) through which will go the shortest paths between vertices in A and C, and we connect a_i to the vertices of V corresponding to the bit-representation of set-rep(i), and c_q (with $i = \phi(q)$) to all the other vertices of V. In this way, the construction will ensure that c_q is covered by a shortest path between $a_{\phi(q)}$ and c_q^b, but is not covered by any other shortest path between a vertex of A and a vertex of C. We give the details in the following subsection.

6.2 Reduction

Consider an instance ψ of 3-Partitioned-3-SAT, with $X^\alpha, X^\beta, X^\gamma$ the partition of the variable set. From ψ, we construct the graph G as follows. We describe the construction of X^α, with the constructions for X^β and X^γ being analogous. See Figure 8 for an illustration. We rename the variables in X^α to x_i^α for $i \in [n]$.

- For every variable x_i^α, we add the vertices t_{2i}^α and f_{2i-1}^α. Formally, $A^\alpha = \{t_{2i}^\alpha, f_{2i-1}^\alpha \mid i \in [n]\}$, and hence, $|A^\alpha| = 2n$.
- For every variable x_i^α, we add four vertices: $x_i^{\alpha,\triangleleft}, x_i^{\alpha,\triangleright}, x_i^{\alpha,\circ}, x_i^{\alpha,\star}$. We make $x_i^{\alpha,\triangleleft}$ and $x_i^{\alpha,\triangleright}$ adjacent to both t_{2i}^α and f_{2i-1}^α. We make $x_i^{\alpha,\circ}$ adjacent to both $x_i^{\alpha,\triangleleft}$ and $x_i^{\alpha,\triangleright}$. We make $x_i^{\alpha,\star}$ adjacent to $x_i^{\alpha,\circ}$.
- We add the vertices y_1, y_2, z_1, z_2. We make y_1 and y_2 adjacent to every vertex of A^α. We make y_i adjacent to z_i for $i \in \{1, 2\}$. Note that y_1, y_2, z_1, z_2 are common to X^β and X^γ.

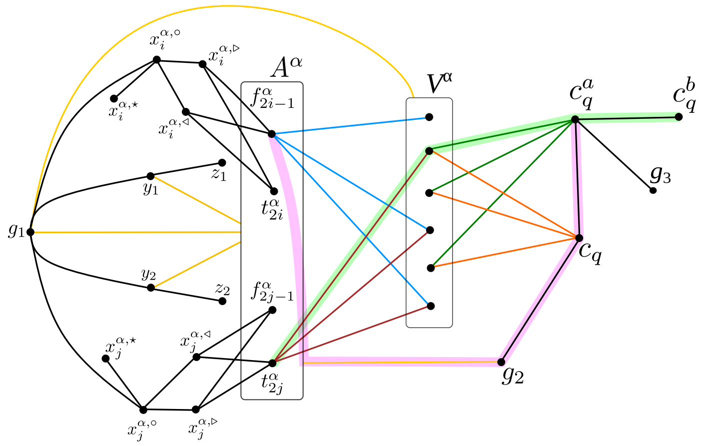

Figure 8 Overview of the reduction. We only draw A^α and V^α here, as A^β, A^γ, V^β, and V^γ are similar. The yellow lines joining g_1, g_2, y_1, and y_2 to sets indicate that the corresponding vertex is adjacent to all the vertices of the corresponding set. Suppose that f_{2i-1}^α and t_{2j}^α are in the geodetic set and $\overline{x}_i$ appears in the clause c_q. The thick green path is a shortest path between t_{2j}^α and c_q^b which does not cover c_q. The thick violet path plus the edge (c_q^a, c_q^b) is a shortest path between f_{2i-1}^α and c_q^b covering c_q.

- We add the vertex g_1 and make it adjacent to y_1, y_2, and $x_i^{\alpha,\circ}$ for each $i \in [n]$. Note that g_1 is common to X^β and X^γ. We add edges between g_1 and every vertex of A^α.
- Let p be the smallest positive integer such that $2n \leq \binom{2p}{p}$. In particular, $p = \mathcal{O}(\log n)$. We add a *validation portal*, a clique on $2p$ vertices, denoted by $V^\alpha = \{v_1^\alpha, v_2^\alpha, \dots, v_{2p}^\alpha\}$. For each $\delta \in \{\alpha, \beta, \gamma\}$, we add edges between g_1 and every vertex of V^δ.
- For every clause C_q in ψ, we introduce three vertices: c_q, c_q^a, c_q^b. We add the edges (c_q, c_q^a) and (c_q^a, c_q^b).
- Define $\textsf{set-rep} : [2n] \mapsto \mathcal{F}_p$ as an arbitrary injective function, where $\mathcal{F}_p$ is the Sperner family (and p is as defined two items above). Add the edge $(t_{2i}^\alpha, v_{p'}^\alpha)$ for every $p' \in \textsf{set-rep}(2i)$ and the edge $(f_{2i-1}^\alpha, v_{p'}^\alpha)$ for every $p' \in \textsf{set-rep}(2i-1)$. If the variable x_i^α appears positively in the clause C_q, then we add the edges $(c_q, v_{p'}^\alpha)$ and $(c_q^a, v_{p'}^\alpha)$ for every $p' \in [2p] \setminus \textsf{set-rep}(2i)$. If the variable x_i^α appears negatively in the clause C_q, then we add the edges $(c_q, v_{p'}^\alpha)$ and $(c_q^a, v_{p'}^\alpha)$ for every $p' \in [2p] \setminus \textsf{set-rep}(2i-1)$.
- Add a vertex g_2 and make g_2 adjacent to every vertex of A^α and every vertex of $\{c_q : q \in [m]\}$. Note that g_2 is common to X^β and X^γ.
- Add a vertex g_3 and make it adjacent to every vertex of $\{c_q^a : q \in [m]\}$. Note that g_3 and the vertices of $\{c_q, c_q^a, c_q^b : q \in [m]\}$ are common to X^β and X^γ.

This concludes the construction of G. The reduction returns (G, k) as an instance of Geodetic Set where $k = 6n + m + 2$.

6.3 Correctness of the Reduction

Suppose, given an instance ψ of 3-Partitioned-3-SAT, that the reduction above returns (G, k) as an instance of Geodetic Set.

▶ **Lemma 12.** *If ψ is a satisfiable 3-Partitioned-3-SAT formula, then G admits a geodetic set of size k.*

Proof. Suppose that $\pi : X^\alpha \cup X^\beta \cup X^\gamma \mapsto \{\texttt{True}, \texttt{False}\}$ is a satisfying assignment for ψ. We construct a geodetic set S of size k for G using this assignment.

For every $\delta \in \{\alpha, \beta, \gamma\}$ and $i \in [n]$, if $\pi(x_i^\delta) = \texttt{True}$, then let $t_{2i}^\delta \in S$, and otherwise, $f_{2i-1}^\delta \in S$. We also put z_1, z_2, $x_i^{\delta,\star}$, and c_q^b into S for all $i \in [n]$, $\delta \in \{\alpha, \beta, \gamma\}$, and $q \in [m]$. Note that $|S| = k$.

Now, we show that S is indeed a geodetic set of G. First, y_1, y_2, z_1, z_2, g_1, and all the vertices of $A^\alpha, A^\beta, A^\gamma$ are covered by a shortest path between z_1 and z_2. Then, for each $\delta \in \{\alpha, \beta, \gamma\}$ and $i \in [n]$, $x_i^{\delta,\triangleleft}$, $x_i^{\delta,\triangleright}$, $x_i^{\delta,\circ}$, and $x_i^{\delta,\star}$ are covered by a shortest path between $S \cap \{t_{2i}^\delta, f_{2i-1}^\delta\}$ and $x_i^{\delta,\star}$. The vertex g_3 is covered by any shortest path between c_q^b and $c_{q'}^b$, where C_q and $C_{q'}$ are two clauses of ψ. Suppose that $\pi(x_i^\delta)$, for some $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$, satisfies some clause C_q. By our construction, if x_i^δ appears positively (negatively, respectively) in C_q, then t_{2i}^δ (f_{2i-1}^δ, respectively) and c_q^b are at distance four since t_{2i}^δ (f_{2i-1}^δ, respectively) and c_q^a have no common neighbor in V^δ. Moreover, there is a shortest path from t_{2i}^δ (f_{2i-1}^δ, respectively) to c_q^b of length four, covering g_2, c_q, c_q^a, and c_q^b; there is also a shortest path from t_{2i}^δ (f_{2i-1}^δ, respectively) to c_q^b of length four, covering $v_j^\delta, v_h^\delta, c_q^a$, and c_q^b, where $v_j^\delta \in V^\delta$ is a vertex adjacent to t_{2i}^δ (f_{2i-1}^δ, respectively) and v_h^δ is any vertex of V^δ that is not adjacent to t_{2i}^δ (f_{2i-1}^δ, respectively). Thus, every vertex of V^δ for $\delta \in \{\alpha, \beta, \gamma\}$ is covered by a shortest path between two vertices of S. Since every clause of ψ is satisfied by π, it follows that every vertex of $\{c_q, c_q^a, c_q^b : q \in [m]\}$ is covered by a shortest path between two vertices of S. As a result, S is a geodetic set of G. ◀

▶ **Lemma 13.** *If G admits a geodetic set of size k, then ψ is a satisfiable* 3-Partitioned-3-SAT *formula.*

Proof. Suppose that G has a geodetic set S of size at most k. By Observation 3, z_1, z_2, $x_i^{\delta,\star}$, and c_q^b for all $i \in [n]$, $\delta \in \{\alpha, \beta, \gamma\}$, and $q \in [m]$ must be in any geodetic set S of G.

▷ Claim 14. For each $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$, exactly one of t_{2i}^δ and f_{2i-1}^δ must be in S.

Proof. Since S is a geodetic set, for each $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$ $x_i^{\delta,\triangleleft}$ and $x_i^{\delta,\triangleright}$ must be covered by shortest paths between two vertices of S. If $t_{2i}^\delta \in S$ ($f_{2i-1}^\delta \in S$, respectively), $x_i^{\delta,\triangleleft}$ and $x_i^{\delta,\triangleright}$ are covered by shortest paths between $t_{2i}^\delta \in S$ ($f_{2i-1}^\delta \in S$, respectively) and $x_i^{\delta,\star}$. Suppose that, for some $i' \in [n]$ and $\delta' \in \{\alpha, \beta, \gamma\}$, neither of $t_{2i'}^{\delta'}$ and $f_{2i'-1}^{\delta'}$ is in S. Moreover, if neither of $x_{i'}^{\delta',\triangleleft}$ and $x_{i'}^{\delta',\triangleright}$ is in S, then, due to the edges incident with g_1, no vertices in S have a shortest path containing any of these two vertices. Similarly, if only one of $x_{i'}^{\delta',\triangleleft}$ and $x_{i'}^{\delta',\triangleright}$ is in S, then the other is not covered by S. Thus, if neither of $t_{2i'}^{\delta'}$ and $f_{2i'-1}^{\delta'}$ is in S, then both $x_{i'}^{\delta',\triangleleft}$ and $x_{i'}^{\delta',\triangleright}$ must be in S. Since $k - |\{z_1, z_2\} \cup \{x_i^{\delta,\star} : i \in [n], \delta \in \{\alpha, \beta, \gamma\}\} \cup \{c_q^b : q \in [m]\}| = 3n$, we conclude that exactly one of t_{2i}^δ and f_{2i-1}^δ must be in S for each $i \in [n]$ and $\delta \in \{\alpha, \beta, \gamma\}$. ◁

By Claim 14 and earlier arguments, we now have that $|S| = k$.

▷ Claim 15. For each $q \in [m]$, the vertex c_q is covered either by a shortest path between c_q^b and t_{2i}^δ, where the variable x_i^δ appears positively in the clause C_q, or by a shortest path between c_q^b and f_{2i-1}^δ, where the variable x_i^δ appears negatively in the clause C_q. Moreover, c_q is covered by no other type of shortest path between two vertices in S.

Proof. By the construction of G, if the variable x_i^δ appears positively in the clause C_q, then there is a shortest path from t_{2i}^δ to c_q^b of length four covering g_2, c_q, c_q^a, and c_q^b. If the variable x_i^δ appears negatively in the clause C_q, then there is a shortest path from f_{2i-1}^δ to c_q^b of length four covering g_2, c_q, c_q^a, and c_q^b.

Next, we show that c_q is not covered by any shortest path between any other two vertices of S. We can check that c_q is not covered by any of the shortest paths between z_1 and z_2, between z_j ($j \in \{1,2\}$) and $x_i^{\delta,\star}$ ($i \in [n], \delta \in \{\alpha,\beta,\gamma\}$), and between z_j ($j \in \{1,2\}$) and $S \cap \{t_{2i}^{\delta}, f_{2i-1}^{\delta}\}$ ($i \in [n], \delta \in \{\alpha,\beta,\gamma\}$). Note that any shortest path from z_j ($j \in \{1,2\}$) to c_q^b ($q \in [m]$) is of length five, covering y_j, some vertex of A^{δ} ($\delta \in \{\alpha,\beta,\gamma\}$), some vertex of V^{δ}, c_q^a, and c_q^b.

We can check that c_q is not covered by any of the shortest paths between $x_i^{\delta,\star}$ and $x_{i'}^{\delta',\star}$ ($i,i' \in [n], \delta,\delta' \in \{\alpha,\beta,\gamma\}$), and between $x_i^{\delta,\star}$ and $S \cap \{t_{2i'}^{\delta'}, f_{2i'-1}^{\delta'}\}$ ($i,i' \in [n], \delta,\delta' \in \{\alpha,\beta,\gamma\}$). Note that any shortest path from $x_i^{\delta,\star}$ ($i \in [n], \delta \in \{\alpha,\beta,\gamma\}$) to c_q^b ($q \in [m]$) is of length five, covering $x_i^{\delta,\circ}$, g_1, some vertex of V^{δ}, c_q^a, and c_q^b.

Note that any shortest path between c_q^b and $c_{q'}^b$ ($q,q' \in [m]$) is of length four, covering c_q^a, g_3, and $c_{q'}^a$.

We can check that c_q is not covered by any shortest paths between $S \cap \{t_{2i}^{\delta}, f_{2i-1}^{\delta}\}$ and $S \cap \{t_{2i'}^{\delta'}, f_{2i'-1}^{\delta'}\}$ ($i,i' \in [n], \delta,\delta' \in \{\alpha,\beta,\gamma\}$).

If the variable x_i^{δ} does not appear positively in the clause C_q, then any shortest path between c_q^b and t_{2i}^{δ} is of length three (because c_q^a and t_{2i}^{δ} have a common neighbour in V^{δ}), covering some vertex of V^{δ} and c_q^a, but not c_q. Similarly, if x_i^{δ} does not appear negatively in C_q, then any shortest path between c_q^b and f_{2i-1}^{δ} is of length three and does not cover c_q.

By the case analysis above, the claim is true. $\lhd$

By Claim 14, exactly one vertex of t_{2i}^{δ} and f_{2i-1}^{δ} belongs to S for each $i \in [n]$ and $\delta \in \{\alpha,\beta,\gamma\}$. We define an assignment π to the variables of ψ as follows. For each $i \in [n]$ and $\delta \in \{\alpha,\beta,\gamma\}$, if $t_{2i}^{\delta} \in S$, then $\pi(x_i^{\delta}) = \texttt{True}$. Otherwise, $\pi(x_i^{\delta}) = \texttt{False}$. Since S is a geodetic set for G, every vertex c_q ($q \in [m]$) is covered by a shortest path between two vertices of S. By Claim 15, every vertex c_q ($q \in [m]$) is covered by a shortest path between $S \cap \{t_{2i}^{\delta}, f_{2i-1}^{\delta}\}$ and c_q^b, where the variable x_i^{δ} appears in the clause C_q. It follows that every clause C_q is satisfied by $\pi(x_i^{\delta})$. As a result, ψ is a satisfiable 3-Partitioned-3-SAT formula. $\blacktriangleleft$

Proof of Theorem 11. First, it is not hard to check that the diameter of G is at most 5. Then, let $X = V^{\alpha} \cup V^{\beta} \cup V^{\gamma} \cup \{g_1, g_2, g_3, y_1, y_2\}$. We can check that every component of $G \setminus X$ has at most six vertices and $|X| = \mathcal{O}(\log n)$. Thus, the treewidth $\texttt{tw}(G)$ — in fact, even the treedepth $\texttt{td}(G)$ — of G is bounded by $\mathcal{O}(\log n)$. By the description of the reduction, it takes polynomial time to compute the reduced instance. Hence, if there is an algorithm for Geodetic Set that runs in time $2^{f(\texttt{diam})^{o(\texttt{tw})}}$ (or $2^{f(\texttt{diam})^{o(td)}}$) then, there is an algorithm running in time $2^{o(n)}$ for 3-Partitioned-3-SAT, which by Proposition 5, contradicts the ETH. $\blacktriangleleft$

7 Strong Metric Dimension: Lower Bound Regarding Vertex Cover

The aim of this section is to prove the following theorem.

▶ **Theorem 16.** *Unless the ETH fails, Strong Metric Dimension does not admit:*

- *an algorithm running in time $2^{2^{o(\texttt{vc})}} \cdot n^{\mathcal{O}(1)}$ for any computable function $f : \mathbb{N} \mapsto \mathbb{N}$, nor*
- *a kernelization algorithm returning a kernel with $2^{o(\texttt{vc})}$ vertices.*

To this end, we present a reduction from Exact-3-Partitioned-3-SAT (see Section 4) to Strong Metric Dimension. We use the relation between Strong Metric Dimension and the Vertex Cover problem, which was established in [85], to prove the theorem. We need the following definition in order to state the relationship.

▶ **Definition 17.** *Given a graph G, we say a vertex $u \in V(G)$ is maximally distant from $v \in V(G)$ if there is no $x \in V(G) \setminus \{u\}$ such that a shortest path between x and v contains u. Formally, for every $y \in N(u)$, we have $d(y, v) \leq d(u, v)$. If u is maximally distant from v, and v is maximally distant from u, then u and v are* mutually maximally distant *in G, and we write $u \bowtie v$.*

For any two mutually maximally distant vertices in G, there is no vertex in G that strongly resolves them, except themselves. Hence, if $u \bowtie v$ in G, then, for any strong resolving set S of G, at least one of u or v is in S, i.e., $|\{u, v\} \cap S| \geq 1$. Oellermann and Peters-Fransen [85] showed that this necessary condition is also sufficient. Consider an auxiliary graph G_{SR} of G defined as follows.

▶ **Definition 18.** *Given a connected graph G, the* strong resolving graph *of G, denoted by G_{SR}, has vertex set $V(G)$ and two vertices u, v are adjacent if and only if u and v are mutually maximally distant in G, i.e., $u \bowtie v$.*

▶ **Proposition 19** (Theorem 2.1 in [85])**.** *For a connected graph G, $\texttt{smd}(G) = \texttt{vc}(G_{SR})$.*

In light of the above proposition, it is sufficient to prove the following lemma.

▶ **Lemma 20.** *There is a polynomial-time reduction that, given an instance ψ of* Exact-3-Partitioned-3-SAT *on $3n$ variables, returns an equivalent instance (G, k) of* Strong Metric Dimension *such that $\texttt{vc}(G) = \mathcal{O}(\log(n))$ and $\texttt{vc}(G_{SR}) = k$.*

Recall the textbook reduction from 3-SAT to Vertex Cover [67]. In this, we add matching edges corresponding to variables, and vertex-disjoint triangles corresponding to clauses. Finally, we add edges between two vertices corresponding to the same literal on the "variable-side" and the "clause-side". We adopt the same reduction for Exact-3-Partitioned 3-SAT to Vertex Cover.

Given an instance ψ of Exact-3-Partitioned-3-SAT, with m clauses and $3n$ variables partitioned equally into $X^\alpha, X^\beta, X^\gamma$, we construct the graph H as follows.

1. We rename the variables in X^α to x_i^α for $i \in [n]$. Analogously, we do this for X^β and X^γ. For every x_i^α, we add two vertices $x_{i,t}^\alpha$ and $x_{i,f}^\alpha$ for the positive and negative literal, respectively. Define X_T^α and X_F^α as the collection of vertices corresponding to the positive and negative literals of the variables in X^α, respectively. We define the sets $X_T^\beta, X_F^\beta, X_T^\gamma$, and X_F^γ, similarly.
   Consider a clause $C_q = (x_i^\alpha \vee \neg x_j^\beta \vee x_\ell^\gamma)$. We add three vertices: $x_{i,t}^{\alpha,q}$, $x_{j,f}^{\beta,q}$, and $x_{\ell,t}^{\gamma,q}$, and the edges to make it a triangle. Let U_T^α be the collection of vertices corresponding to the positive literals of variables in X^α. Formally, $U_T^\alpha = \{x_{i,t}^{\alpha,q} \mid$ there is a clause C_q that contains the positive literal of the variable $x_i^\alpha\}$. Note that, for a variable x_i^α, there may be multiple vertices corresponding to its positive literal in U_T^α depending on the number of clauses that contain x_i^α. We analogously define U_F^α, U_T^β, U_F^β, U_T^γ, and U_F^γ. See Figure 9 for an illustration.
2. We add matching edges across X_T^α and X_F^α connecting $x_{i,t}^\alpha$ and $x_{i,f}^\alpha$ for every $i \in [n]$. We add similar matching edges across X_T^β and X_F^β, and X_T^γ and X_F^γ. See the green edges in Figure 9.
3. As ψ is an instance of Exact-3-Partitioned-3-SAT, for any clause there is a triangle that contains exactly one vertex from each of $U_T^\alpha \cup U_F^\alpha$, $U_T^\beta \cup U_F^\beta$, and $U_T^\gamma \cup U_F^\gamma$. See the purple triangle in Figure 9. For each clause, we add the edges to form its corresponding triangle.

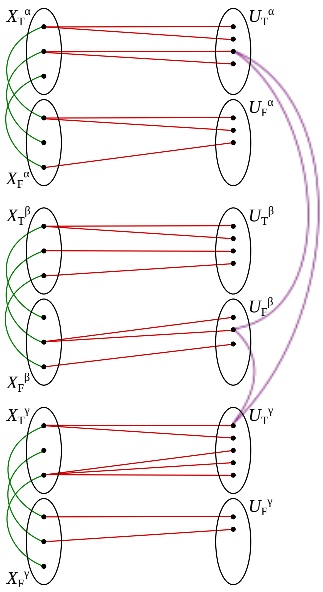

Figure 9 Overview of the reduction from EXACT-3-PARTITIONED-3-SAT to VERTEX COVER. In Step 1, we add all the independent sets mentioned. In Step 2, we add matching green edges corresponding to the assignment of the variables. In Step 3, we add purple triangles corresponding to clauses. For example, the purple triangle corresponds to the clause $(x_3^\alpha \vee \neg x_2^\beta \vee x_1^\gamma)$. In Step 4, we add red edges connecting the vertices corresponding to the same literal on the "variable-side" and "clause-side".

4. Finally, we add edges connecting a vertex corresponding to a literal on the variable-side to vertices corresponding to the same literal on the clause-side. See the red edges in Figure 9.

The reduction returns (H, k) as the reduced instance of VERTEX COVER, where $k = 3n + 2m$. The correctness of the following lemma is spelled out in [67].

▶ **Lemma 21.** *The formula ψ, with $3n$ variables and m clauses, is a satisfiable EXACT-3-PARTITIONED-3-SAT formula if and only if H admits a vertex cover of size $k = 3n + 2m$.*

In what remains of this section, our objective is to construct G such that G_{SR} is as "close" to H as possible. It will be helpful to think about $V(H)$ as a subset of $V(G) = V(G_{SR})$. We want to construct G such that all the edges in $E(H)$ are present in G_{SR}, while no undesirable edge appears in G_{SR}. We use a *set representation gadget* and a *bit representation gadget* for the first and the second part, respectively. However, for the second part, we need another trick since we may have some undesirable edges. We ensure that all the edges in $E(G_{SR}) \setminus E(H)$, i.e., undesirable edges, are incident to a clique, say Z, in G_{SR}. Moreover, there is a vertex z in Z such that $N[z] = Z$, i.e., z is not adjacent to any vertex in $V(G_{SR}) \setminus Z$. Then, without loss of generality, we can assume that any vertex cover of G_{SR} contains $Z \setminus \{z\}$. Hence, all the undesired edges are deleted by a pre-processing step while finding the vertex cover of G_{SR}. In other words, $E(H) = E(G_{SR} - (Z \setminus \{z\}))$. This ensures that the difficulty of finding a strong resolving set in G is encoded in finding a vertex cover in $G_{SR} - Z$, a graph with only desired edges. We note the following easy observation before presenting the primary tools used in the reduction.

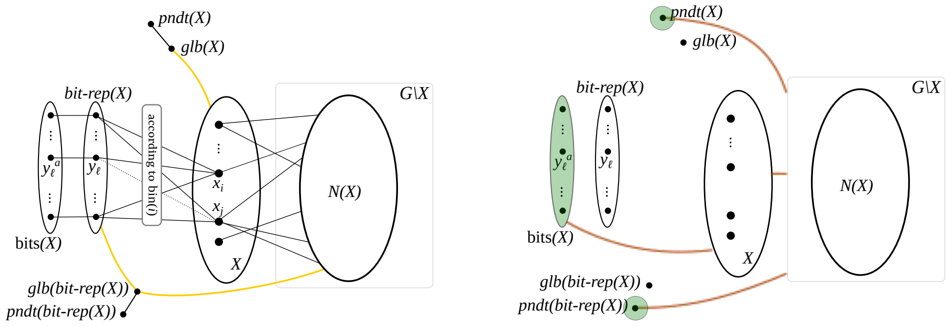

Figure 10 Set Identifying Gadget. The graph G is on the left side, and the corresponding G_{SR} is on the right side. The green-shaded region in G_{SR} denotes a clique. Note that the brown edges in G_{SR} are not relevant at this time.

▶ **Observation 22.** *Consider a connected graph G that has at least 3 vertices. Suppose $Z \subseteq V(G)$ is the collection of all the pendent vertices in G. Then, Z is a clique in G_{SR}, and every vertex in $N(Z)$ is an isolated vertex in G_{SR}.*

Proof. Consider any two vertices z_1, z_2 in Z, and let u_2 be the unique neighbor of z_2. It is easy to see that $d(z_1, u_2) < d(z_1, z_2)$. Hence, z_2 is maximally distant from z_1. Using the symmetric arguments, z_1 is maximally distant from z_2, and hence, $z_1 \bowtie z_2$. By the construction of G_{SR}, there is an edge with endpoints z_1, z_2. As these were two arbitrary points in Z, we have that Z is a clique in G_{SR}.

Consider an arbitrary vertex $v \in V(G) \setminus \{u_2\}$. If $v \neq z_2$, then $d(v, z_2) > d(v, u_2)$ and hence, u_2 is *not* maximally distant from v. This implies that u_2 is not adjacent with any vertex in $V(G_{SR}) \setminus \{z_2\}$ in G_{SR}. As G is connected and has at least three vertices, u_2 is not maximally distant from z_2 either. Hence, u_2 is an isolated vertex in G_{SR}. Since u_2 is an arbitrary vertex in $N(Z)$, the second part of the claim follows. ◀

7.1 Preliminary Tools

7.1.1 Bit Representation Gadget to Add Independent Sets

In this subsection, we accomplish Step 1 of the reduction mentioned at the start of the section. Formally, given a graph G' and an independent set $X \subseteq V(G')$ of its vertices, we want to add vertices and edges to G' to obtain a graph G such that X remains an independent set in G, and X is also an independent set in G_{SR}. We do this as follows:

- First, let $X = \{x_i \mid i \in [|X|]\}$, and set $q := \lceil \log(|X| + 1) \rceil$. We select this value for q to uniquely represent each integer in $[|X|]$ by its bit-representation in binary (note that we start from 1 and not 0).
- For every $\ell \in [q]$, add two vertices: y^a_ℓ and y_ℓ, and the edge (y^a_ℓ, y_ℓ). We denote $\mathsf{bit\text{-}rep}(X) = \{y_\ell \mid \ell \in [q]\}$ and $\mathsf{bits}(X) = \{y^a_\ell \mid \ell \in [q]\}$ for convenience in a later case analysis. Note that both $\mathsf{bit\text{-}rep}(X)$ and $\mathsf{bits}(X)$ are independent sets and $\mathsf{bits}(X)$ is a collection of pendent vertices in G whose neighborhoods are in $\mathsf{bit\text{-}rep}(X)$.
- For every integer $j \in [|X|]$, let $\mathsf{bin}(j)$ denote the binary representation of j using q bits. Connect x_j with y_i if the i^{th} bit (going from left to right) in $\mathsf{bin}(j)$ is 1.
- Add a vertex, denoted by $\mathsf{glb}(X)$, and make it adjacent to every vertex in X. Add another vertex, denoted by $\mathsf{pndt}(X)$, and make it adjacent only to $\mathsf{glb}(X)$.
- Similarly, add a vertex $\mathsf{glb}(\mathsf{bit\text{-}rep}(X))$ which is adjacent to every vertex in $\mathsf{bit\text{-}rep}(X)$, and a vertex $\mathsf{pndt}(\mathsf{bit\text{-}rep}(X))$ which is adjacent only to $\mathsf{glb}(\mathsf{bit\text{-}rep}(X))$.

This completes the construction of G. We use $\mathsf{glb}(\cdot)$ and $\mathsf{pndt}(\cdot)$ as a function to denote vertices that are adjacent to every vertex in the set, i.e., global to the set, and the vertex pendent to the global vertex of the set, respectively. We mention the small caveat in this notation. Pendent vertices in $\mathsf{bits}(X) \cup \{\mathsf{pndt}(X)\}$ are *not* adjacent to any vertex in X. It is helpful to think of these pendent vertices together with $\mathsf{bit\text{-}rep}(X) \cup \{\mathsf{glb}(X), \mathsf{pndt}(\mathsf{bit\text{-}rep}(X)), \mathsf{glb}(\mathsf{bit\text{-}rep}(X))\}$ as vertices added for X. Moreover, at a later stage in the reduction, we may make $\mathsf{glb}(X)$ global to every vertex in a new set Y. However, we do *not* rename it to $\mathsf{glb}(X \cup Y)$ for notational clarity. Note that no vertex in $\mathsf{bit\text{-}rep}(X) \cup \mathsf{bits}(X)$ is adjacent to any vertex in $V(G) \setminus (\mathsf{bit\text{-}rep}(X) \cup \mathsf{bits}(X) \cup X \cup \mathsf{glb}(\mathsf{bit\text{-}rep}(X))$. See Figure 10 for an illustration.

We use this gadget as a building block for our reduction and do not add any more edges whose both endpoints are completely inside the gadget. While adding other vertices and edges, we ensure that we do not add a vertex whose neighborhood intersects both X and $\mathsf{bit\text{-}rep}(X)$. We now show that the following property (we aimed for) holds under this condition.

▶ **Lemma 23.** *Consider the graph G, an independent set X, and the vertices and edges as defined above. Suppose there is no vertex $v \in V(G)$ that is adjacent to both a vertex in X and a vertex in $\mathsf{bit\text{-}rep}(X)$. Then, X is an independent set in G_{SR}.*

Proof. Consider any two vertices, say x_i, x_j, in X. We want to prove that these two vertices are not adjacent in G_{SR}. By the construction, it is sufficient to prove that either x_i is not maximally disjoint from x_j, or x_j is not maximally disjoint from x_i. As the bit-representation of i is not the same as j, there is a vertex, say y_ℓ in $\mathsf{bit\text{-}rep}(X)$, that is adjacent to x_j, but not to x_i (or vice-versa). We consider the first case. Note that $d(x_i, x_j) = 2$ since $\mathsf{glb}(X)$ is adjacent to both x_i and x_j, and X is an independent set. Since, there is no vertex $v \in V(G)$ such that $N(v) \cap X \neq \emptyset$ and $N(v) \cap \mathsf{bit\text{-}rep}(X) \neq \emptyset$, and $\mathsf{bit\text{-}rep}(X)$ is an independent set, we have $d(x_i, y_\ell) > 2$. Thus, $d(x_i, y_\ell) = 3$. Hence, there exists a vertex in $N(x_j)$ which is farther from x_i. This implies that x_j is not maximally distant from x_i, concluding the proof. ◀

7.1.2 Set Representation Gadget to Add Edges

We use a set representation gadget to add edges across two independent sets in G_{SR}. This will be useful to accomplish Steps 2, 3, and 4.

Consider two independent sets A and B in G', and suppose there is a function $\phi : B \mapsto A$. This function may not be one-to-one. As we are defining this function, it will be helpful to consider A, B as an ordered pair. Our objective is to add vertices and edges to G' to obtain G such that G_{SR} contains an edge (a_i, b_j) for some $a_i \in A$ and $b_j \in B$ if and only if $a_i = \phi(b_j)$. Moreover, we want A, B to remain as independent sets in G and G_{SR}.

We change the function ϕ as per our requirements. In Step 2, we need to add the edge between $x^\alpha_{i,t} \in X^\alpha_T$ and $x^\alpha_{i,f} \in X^\alpha_F$, and other such pairs. In this case, we define $\phi(x^\alpha_{i,f}) = x^\alpha_{i,t}$. Here, ϕ is a one-to-one and onto function from B to A. Consider a clause $C_q = (x^\alpha_i \vee \neg x^\beta_j \vee x^\gamma_\ell)$. As mentioned before, in the reduction, we add the vertices $x^{\alpha,q}_{i,t}$ to U^α_T, $x^{\beta,q}_{j,f}$ to U^β_T, and $x^{\gamma,q}_{\ell,t}$ to U^γ_T. We expect a triangle with these three vertices. Hence, in Step 3, while adding edges across U^α_T and U^β_F, we define the function ϕ as $\phi(x^{\beta,q'}_{j,f}) = x^{\alpha,q}_{i,t}$ if and only if $q = q'$, i.e., if the literals corresponding to these two vertices appear in a clause. For Step 4, while adding edges across X^α_T and U^α_T, define $\phi(x^{\alpha,q'}_{i',t}) = x^\alpha_{i,t}$ if and only if $i = i'$, i.e., if these two vertices correspond to the same literal. In this case, multiple vertices in U^α_T may be assigned to a single vertex in X^α_T if they correspond to the same literal in X^α_T. For example, $\phi(x^{\alpha,q}_{i,t}) = \phi(x^{\alpha,q'}_{i,t}) = x^\alpha_{i,t}$.

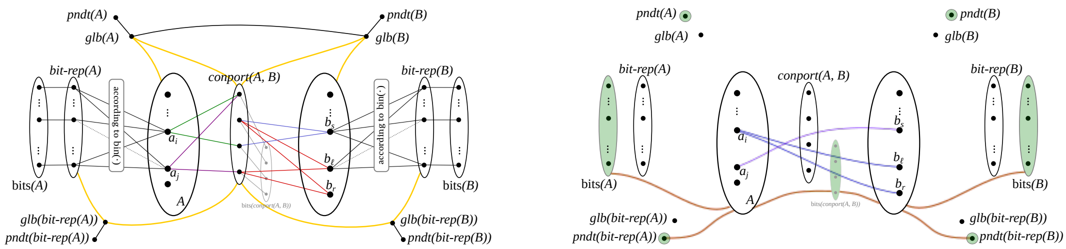

Figure 11 Set Representation Gadget to add edges across independent sets A and B. The graph G is on the left and the graph G_{SR} is on the right. The yellow lines from a vertex to a set represent that the vertex is adjacent to every vertex in the set. Suppose that $\phi(b_\ell) = \phi(b_r) = a_i$ and $\phi(b_s) = a_j$. Note that in G, due to ϕ, a_i shares no common neighbor in $\mathsf{con\text{-}port}(A, B)$ with b_ℓ nor b_r, and that a_j shares no common neighbor in $\mathsf{con\text{-}port}(A, B)$ with b_s. Furthermore, in G, we have $a_i \bowtie b_\ell$, $a_i \bowtie b_r$, and $a_j \bowtie b_s$, which create the edges between these pairs of vertices in G_{SR}. Lastly, the green-shaded region in G_{SR} denotes a clique.

We use *set representations* of integers to achieve this objective and recall some ideas from Section 5.1.4. For a positive integer p, define $\mathcal{F}_p$ as the collection of subsets of $[2p]$ that contains exactly p integers. We critically use the fact that no set in $\mathcal{F}_p$ is contained in any other set in $\mathcal{F}_p$ (such a collection of sets are called a *Sperner family*). This implies for any two different sets $A, B \in \mathcal{F}_p$, A intersects the complement of B, i.e., $A \cap ([2p] \setminus B) \neq \emptyset$. Let n be a positive integer such that $n \leq \binom{2p}{p}$. We define $\mathsf{set\text{-}rep} : [n] \mapsto \mathcal{F}_p$ as a one-to-one function by arbitrarily assigning a set in $\mathcal{F}_p$ to an integer in $[n]$. By the asymptotic estimation of the central binomial coefficient, $\binom{2p}{p} \sim \frac{4^p}{\sqrt{\pi \cdot p}}$ [59]. To get the upper bound of p, we scale down the asymptotic function and have $n \leq \frac{4^p}{2^p} = 2^p$. Thus, $p = \mathcal{O}(\log n)$. Given independent sets $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_{n'}\}$ in G', where $n' \leq n$ and the function $\phi : B \mapsto A$, we add vertices and edges to G' to obtain G as follows.

- We add the sets of vertices $\mathsf{bit\text{-}rep}(A)$, $\mathsf{bits}(A)$, the vertices $\mathsf{glb}(A)$, $\mathsf{pndt}(A)$, $\mathsf{glb}(\mathsf{bit\text{-}rep}(A))$, $\mathsf{pndt}(\mathsf{bit\text{-}rep}(A))$, and the appropriate edges as mentioned in the previous subsection. Similarly, we add the corresponding vertices and edges with respect to B.
- We add the edge $(\mathsf{glb}(A), \mathsf{glb}(B))$.
- We add a *connection portal*, denoted by $\mathsf{con\text{-}port}(A, B) = \{v_1, v_2, \ldots, v_{2p}\}$. For every vertex $v_{p'}$ in $\mathsf{con\text{-}port}(A, B)$, we add a new vertex and make it adjacent to $v_{p'}$. The collection of these pendent vertices are denoted by $\mathsf{bits}(\mathsf{con\text{-}port}(A, B))$.
- For every $i \in [n]$ and for every $p' \in \mathsf{set\text{-}rep}(i)$, we add the edge $(a_i, v_{p'})$. If $\phi(b_j) = a_i$ for some $j \in [n']$, then we add the edge $(b_j, v_{p'})$ for every p' *not* in $\mathsf{set\text{-}rep}(i)$. This ensures that, for every pair a_i, b_j, if $a_i = \phi(b_j)$, then there is no vertex in $\mathsf{con\text{-}port}(A, B)$ that is adjacent to both a_i and b_j.
- Finally, we make every vertex in $\mathsf{con\text{-}port}(A, B)$ adjacent to $\mathsf{glb}(A)$, $\mathsf{glb}(\mathsf{bit\text{-}rep}(A))$, $\mathsf{glb}(B)$, and $\mathsf{glb}(\mathsf{bit\text{-}rep}(B))$.

This completes the construction of G. See Figure 11 for an illustration. As in the previous case, we use this gadget as a building block for our reduction and do not add any more edges whose both endpoints are completely inside the gadget. While adding other vertices and edges, we ensure that we do not add another vertex (outside $\mathsf{con\text{-}port}(A, B)$) whose neighbourhood intersects both A and B. We now show that the following property (we aimed for) holds under this condition.

▶ **Lemma 24.** *Consider the graph G, independent sets A, B, connection portal $\mathsf{con\text{-}port}(A, B)$ added with respect to the function $\phi : B \mapsto A$, and the vertices and edges as defined above.*

For every vertex $v \in V(G) \setminus (\textsf{con-port}(A, B))$, suppose the following conditions are true.
- *It is* not *adjacent to both a vertex in A and a vertex in B;*
- *it is* not *adjacent to both a vertex in B and a vertex in $\textsf{bit-rep}(A)$;*
- *it is* not *adjacent to both a vertex in A and a vertex in $\textsf{bit-rep}(B)$.*

Then, the edge (a_i, b_j) is present in G_{SR} if and only if $a_i = \phi(b_j)$.

Proof. Consider the vertices $a_i, a_j \in A$ and $b_\ell, b_r, b_s \in B$ such that $\phi(b_\ell) = \phi(b_r) = a_i$ and $\phi(b_s) = a_j$ (see Figure 11). We focus on the vertex a_i. Note that every vertex in $\textsf{con-port}(A, B)$ is at distance 1 (if its in $N(a_i)$) or at distance 2 (as $\textsf{glb}(A)$ is adjacent to every vertex in $A \cup \textsf{con-port}(A, B)$) from a_i. By the construction, the vertex $\textsf{glb}(\textsf{bit-rep}(B))$ is at distance 2 from a_i, and hence, every vertex in $\textsf{bit-rep}(B)$ is at distance at most 3 from a_i. Because of the edge $(\textsf{glb}(A), \textsf{glb}(B))$ and the global nature of the endpoints of this edge, every vertex in B is at distance at most 3 from a_i. Since $\phi(b_\ell) = a_i$, as mentioned before, there is no vertex in $\textsf{con-port}(A, B)$ that is adjacent to both a_i and b_ℓ. Moreover, there is no vertex v in $V(G) \setminus \textsf{con-port}(A, B)$, such that $N(v) \cap A \neq \emptyset$ and $N(v) \cap B \neq \emptyset$. Hence, by the construction, b_ℓ is at distance 3 from a_i. Hence, for every vertex $x \in N(b_\ell)$, we have $d(a_i, x) \leq d(a_i, b_\ell)$. This implies that b_ℓ is maximally distant from a_i. As the gadget constructed is symmetric, it is easy to see that a_i is also maximally distant from b_ℓ. Hence, we have $a_i \bowtie b_\ell$ and (a_i, b_ℓ) is an edge in G_{SR}. Using similar arguments, the edges (a_i, b_r) and (a_j, b_s) are present in G_{SR}.

Now, consider $b_s \in B$ and $a_i \in A$ such that $\phi(b_s) \neq a_i$. Then, by the properties of the set representation gadget, there exists a vertex in $\textsf{con-port}(A, B)$ that is adjacent to both a_i and b_s. Hence, $d(a_i, b_s) = 2$. The neighbors of b_s in $\textsf{bit-rep}(B)$ are at distance at most 3 from a_i. However, as there is no vertex $v \in V(G)$ such that $N(v) \cap A \neq \emptyset$ and $N(v) \cap \textsf{bit-rep}(B) \neq \emptyset$, every neighbor of b_s in $\textsf{bit-rep}(B)$ is at distance exactly 3 from a_i. Hence, b_s is not maximally distant from a_i. This implies that there is no edge with endpoints a_i, b_s in G_{SR}. Hence, the edge $(a_{i'}, b_{j'})$ is present in G_{SR} if and only if $\phi(b_{j'}) = a_{i'}$. ◀

Before presenting the reduction, we note that there is a different schematic representation of the gadgets mentioned above in Figure 12.

7.2 Reduction

Consider an instance ψ of Exact-3-Partitioned-3-SAT with m clauses and $3n$ vertices that are partitioned into $X^\alpha, X^\beta, X^\gamma$. Recall the reduction mentioned before Lemma 21. The reduction we present here, constructs a graph G using the construction specified earlier, according to the steps mentioned in the reduction.

- Recall the sets defined in Step 1. For every $A \in \{X^\delta_T, X^\delta_F, U^\delta_T, U^\delta_F \mid \delta \in \{\alpha, \beta, \gamma\}\}$, we add the sets A, $\textsf{bit-rep}(A)$, $\textsf{bits}(A)$, the vertices $\textsf{glb}(A)$, $\textsf{pndt}(A)$, $\textsf{glb}(\textsf{bit-rep}(A))$, and $\textsf{pndt}(\textsf{bit-rep}(A))$, and the associated edges as mentioned in Section 7.1.1.
- For the edges mentioned in Step 2, for every $\delta \in \{\alpha, \beta, \gamma\}$, we assign $A = X^\delta_T$ and $B = X^\delta_F$, and define $\phi : B \mapsto A$ as $\phi(x^\delta_{i,f}) = x^\delta_{i,t}$. Then, we add the connection portal $\textsf{con-port}(X^\delta_T, X^\delta_F)$ and the other vertices and edges mentioned in Section 7.1.2.
- For the edges mentioned in Step 3, for $\delta \in \{\alpha, \beta, \gamma\}$, define $U^\delta = U^\delta_T \cup U^\delta_F$. First, we assign $A = U^\alpha$ and $B = U^\beta$, and define $\phi : B \mapsto A$ as follows: for every clause $C_q = (x^\alpha_i \vee \neg x^\beta_j \vee x^\gamma_\ell)$, $\phi(x^{\beta,q}_{j,f}) = x^{\alpha,q}_{i,t}$. We add $\textsf{con-port}(U^\alpha, U^\beta)$ and the corresponding vertices and edges as specified in Section 7.1.2. We let $\textsf{bit-rep}(U^\alpha) = \textsf{bit-rep}(U^\alpha_T) \cup \textsf{bit-rep}(U^\alpha_F)$ and $\textsf{bit-rep}(U^\beta) = \textsf{bit-rep}(U^\beta_T) \cup \textsf{bit-rep}(U^\beta_F)$. We repeat the process for the pairs (U^β, U^γ) and (U^γ, U^α). As ψ is an instance of Exact-3-Partitioned-3-SAT, every clause contains exactly three variables, and hence, in each case, ϕ is well-defined. We remark that, for

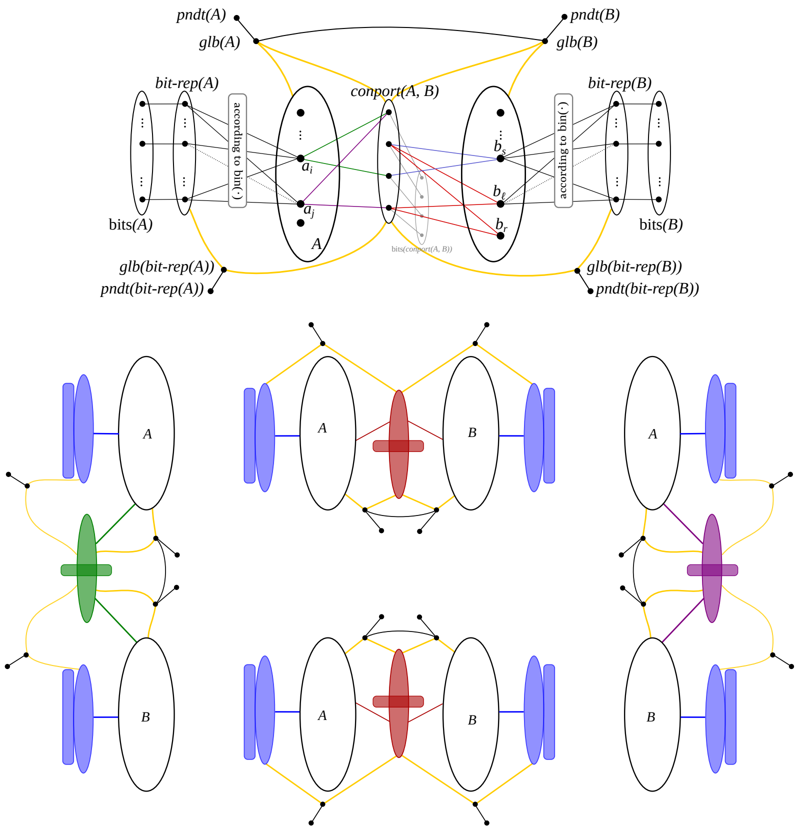

Figure 12 Set Representation Gadget to add edges across independent sets A and B, and four different schematic representations of the same below. In the schematic representation, yellow thick edges denote that the vertex is adjacent with every vertex in the set. The blue filled oval shape corresponds to bit-rep$(\cdot)$, and the blue thick lines denote that the set, say A, is connected to bit-rep(A) according to its bit representation as mentioned in Section 7.1.1. The filled oval shape of green, red, or purple color denotes con-port$(\cdot,\cdot)$. The filled rectangle shape denotes bits$(\cdot)$. The colors of connection ports correspond to the edges mentioned in Figure 9.

every U^δ, there are two global vertices now. We denote them by glb(U^δ) and glb$^\circ(U^\delta)$. However, we do *not* add another bit-rep(U^δ).

Figure 13 shows the vertices and edges added so far.

- For the edges mentioned in Step 4, we first consider $A = X_T^\alpha$ and $B = U_T^\alpha$, and define the function $\phi : B \mapsto A$ as follows: for every $x_{i,t}^{\alpha,q}$ in U^α for some $q \in [m]$ and $i \in [n]$, define $\phi(x_{i,t}^{\alpha,q}) = x_{i,t}^{\alpha}$. We add con-port$(X_T^\alpha, U_T^\alpha)$ and the corresponding vertices and edges as specified in Section 7.1.2. However, the sets bit-rep(X_T^α) and bit-rep(U_T^α), and the vertices glb(X_T^α) and glb(U_T^α) are already defined. Hence, we reuse the sets and introduce some new vertices.
  - We add the vertex glb$^\star(X_T^\alpha)$ and make it adjacent to every vertex in X_T^α. We also add pndt$^\star(X_T^\alpha)$ which is adjacent to only glb$^\star(X_T^\alpha)$. Similarly, we add glb$^\star(U_T^\alpha)$, pndt$^\star(U_T^\alpha)$, and the corresponding edges.
  - We add the vertex glb*(bit-rep(X_T^α)) and make it adjacent to every vertex in bit-rep(X_T^α). We also add pndt*(bit-rep(X_T^α)) which is adjacent to only glb*(bit-rep(X_T^α)). Similarly, we add glb*(bit-rep(U_T^α)), pndt*(bit-rep(U_T^α)), and the corresponding edges.
  - We add con-port(X_T^α, U_T^α) and bits(con-port(X_T^α, U_T^α)) as mentioned in Section 7.1.2.

- Finally, we make every vertex in $\mathsf{con\text{-}port}(X_T^\alpha, U_T^\alpha)$ adjacent to $\mathsf{glb}^\star(X_T^\alpha)$, $\mathsf{glb}^\star(\mathsf{bit\text{-}rep}(X_T^\alpha))$, $\mathsf{glb}^\star(U_T^\alpha)$, and $\mathsf{glb}^\star(\mathsf{bit\text{-}rep}(U_T^\alpha))$.

See Figure 13. We repeat the process for the remaining pairs (X_T^δ, U_T^δ) and (X_F^δ, U_F^δ) for $\delta \in \{\alpha, \beta, \gamma\}$.

- In the final step, consider the following twelve vertices: $\mathsf{glb}^\star(X_T^\delta)$, $\mathsf{glb}^\star(X_F^\delta)$, $\mathsf{glb}^\star(U_T^\delta)$, and $\mathsf{glb}^\star(U_F^\delta)$ for $\delta \in \{\alpha, \beta, \gamma\}$. These vertices are highlighted using filled orange circles around them in Figure 14. We add the edges to convert these vertices into a clique (omitted in Figure 14). We denote it by S_K, for *short-cut clique*, as it provides all the necessary short-cuts. We remark that such additional edges are not necessary for the other $\mathsf{glb}(\cdot)$ vertices.

This completes the construction of G.

Suppose Z is the collection of all the pendent vertices in G. Formally,

$$
\begin{aligned}
Z = & \left[\bigcup_{\delta \in \{\alpha,\beta,\gamma\}} \mathsf{bits}(X_T^\delta) \cup \mathsf{bits}(X_F^\delta) \cup \mathsf{bits}(U_T^\delta) \cup \mathsf{bits}(U_F^\delta)\right] \bigcup \\
& \left[\bigcup_{\delta \in \{\alpha,\beta,\gamma\}} \mathsf{bits}(\mathsf{con\text{-}port}(X_T^\delta, X_F^\delta))\right] \bigcup \left[\bigcup_{\delta \neq \epsilon \in \{\alpha,\beta,\gamma\}} \mathsf{bits}(\mathsf{con\text{-}port}(U^\delta, U^\epsilon))\right] \bigcup \\
& \left[\bigcup_{\delta \in \{\alpha,\beta,\gamma\}} \mathsf{bits}(\mathsf{con\text{-}port}(X_T^\delta, U_T^\delta)) \cup \mathsf{bits}(\mathsf{con\text{-}port}(X_F^\delta, U_F^\delta))\right] \bigcup \\
& \left\{\mathsf{pndt}(X_T^\delta), \mathsf{pndt}(X_F^\delta), \mathsf{pndt}^\star(X_T^\delta), \mathsf{pndt}^\star(X_F^\delta), \mid \delta \in \{\alpha,\beta,\gamma\}\right\} \cup \\
& \left\{\mathsf{pndt}(\mathsf{bit\text{-}rep}(X_T^\delta)), \mathsf{pndt}(\mathsf{bit\text{-}rep}(X_F^\delta)), \mathsf{pndt}^\star(\mathsf{bit\text{-}rep}(X_T^\delta)), \mathsf{pndt}^\star(\mathsf{bit\text{-}rep}(X_F^\delta)) \mid \delta \in \{\alpha,\beta,\gamma\}\right\} \\
& \cup \left\{\mathsf{pndt}^\star(U_T^\delta), \mathsf{pndt}^\star(U_F^\delta), \mathsf{pndt}^\star(\mathsf{bit\text{-}rep}(U_T^\delta)), \mathsf{pndt}^\star(\mathsf{bit\text{-}rep}(U_F^\delta)), \mid \delta \in \{\alpha,\beta,\gamma\}\right\} \bigcup \\
& \cup \left\{\mathsf{pndt}(U^\delta), \mathsf{pndt}^\circ(U^\delta), \mathsf{pndt}(\mathsf{bit\text{-}rep}(U^\delta)), \mathsf{pndt}^\circ(\mathsf{bit\text{-}rep}(U^\delta)) \mid \delta \in \{\alpha,\beta,\gamma\}\right\} \bigcup
\end{aligned}
$$

The reduction returns (G, k) as an instance of Strong Metric Dimension, where $k = 3n + 2m + (|Z| - 1)$.

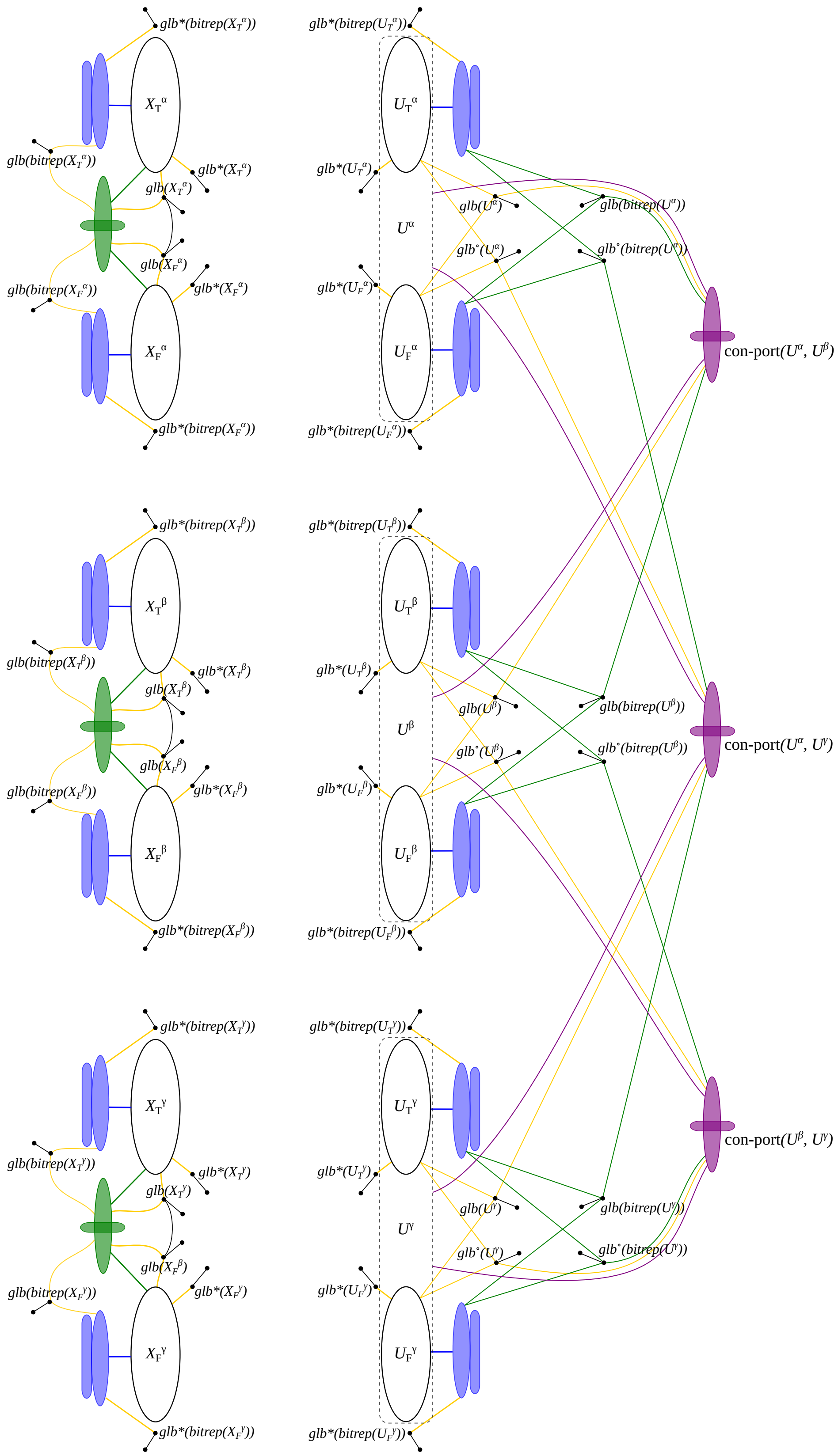

Figure 13 Overview of the vertices and edges added in the first step of the reduction along with some new vertices like $\mathsf{glb}^\star(\cdot)$, which we define soon. Please refer to Figure 15 for a more streamlined illustration of connections on clause-side vertices. We highlight that the construction so far is replicating the gadget mentioned in Subsection 7.1.1 and 7.1.2. Hence, it also satisfies the premises of Lemma 23 and 24. This implies these vertices and edges across them are identical to corresponding vertices and edges in H.

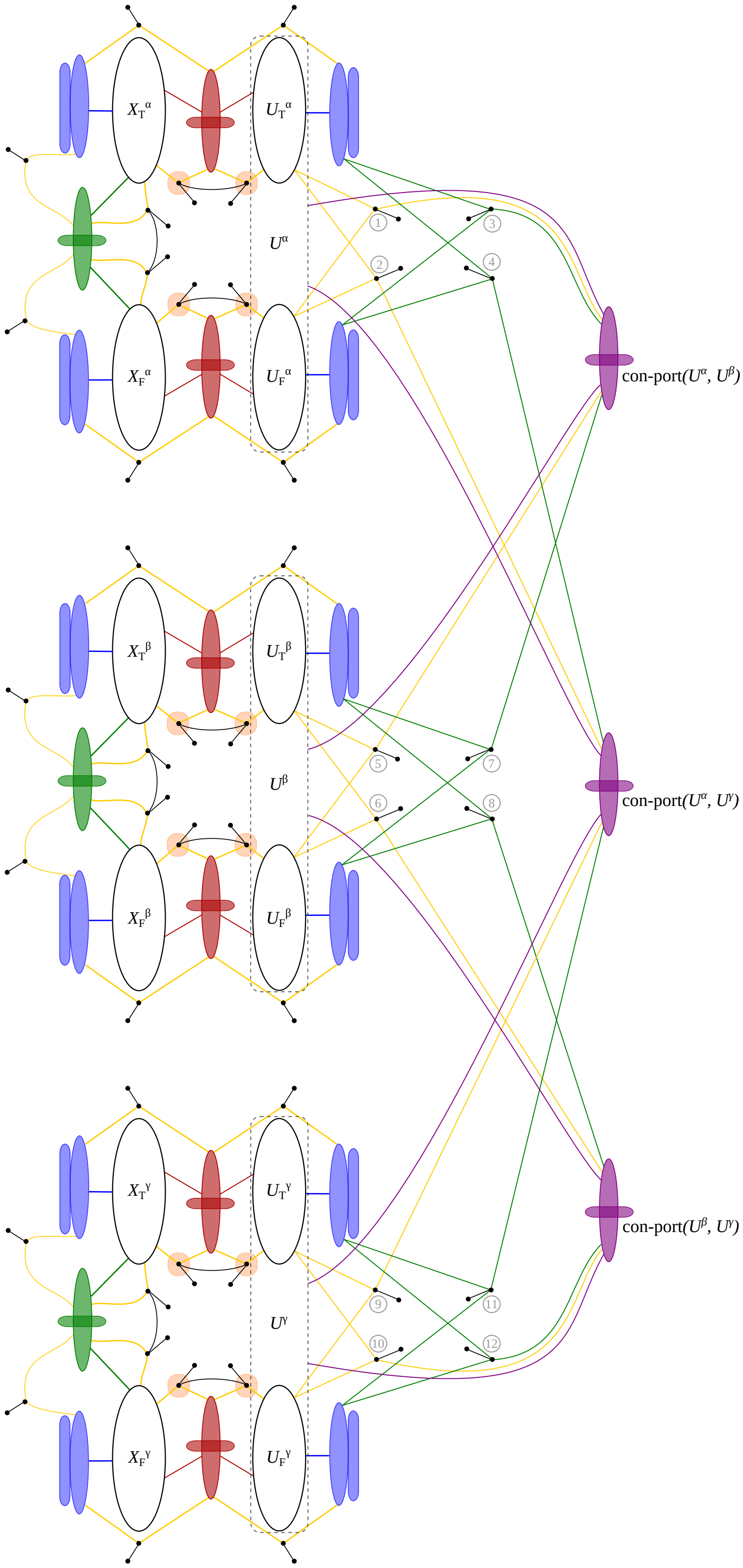

Figure 14 Overview of the reduction. The orange circled vertices denote the short-cut clique S_K. The edges $(1, 5)$, $(3, 9)$, and $(6, 10)$ are not shown to preserve clarity.

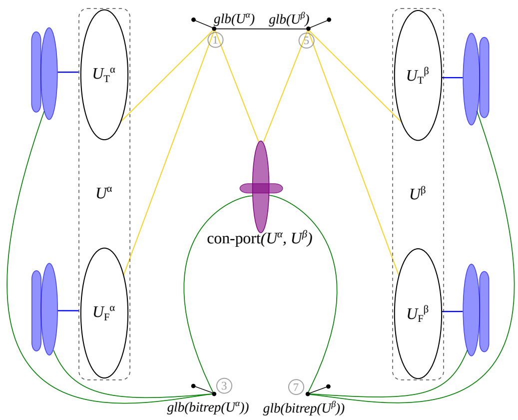

Figure 15 Highlighting the connections on the clause side.

7.3 Correctness of the Reduction

Suppose, given an instance ψ of Exact-3-Partitioned-3-SAT, that the reduction above returns (G, k) as an instance of Strong Metric Dimension.

▶ **Lemma 25.** *ψ is a satisfiable Exact-3-Partitioned-3-SAT formula if and only if G admits a strong resolving set of size k.*

Proof. We note that the construction is very symmetric with respect to the partition X^α, X^β, X^γ. Recall that, from the graph G, we construct G_{SR} as mentioned in Definition 18. Hence, $V(G) = V(G_{SR})$. Moreover, by Proposition 19, $\mathtt{smd}(G) = \mathtt{vc}(G_{SR})$. Hence, it is sufficient to prove that ψ is a satisfiable Exact-3-Partitioned-3-SAT formula if and only if G_{SR} admits a vertex cover of size k. We start by identifying all the edges in G_{SR}.

Define the subsets X an U of $V(G)$ as $X := \bigcup_{\delta \in \{\alpha,\beta,\gamma\}} (X_T^\delta \cup X_F^\delta)$ and $U := \bigcup_{\delta \in \{\alpha,\beta,\gamma\}} (U_T^\delta \cup U_F^\delta)$. Observe that $\langle Z, N(Z), X, U\rangle$ is a partition of $V(G)$. By Observation 22, $N(Z)$ is a collection of pendent vertices in G_{SR}, and hence, their presence in G_{SR} is irrelevant while computing its vertex cover. Hence, we need to focus on Z, X, U, and the edges across and within these sets in G_{SR}. By Observation 22, Z is a clique in G_{SR}. We first prove that there is a vertex $z \in Z$ such that z is not adjacent to any vertex in $X \cup U$. Hence, it is safe to assume that any vertex cover of G_{SR} contains $Z \setminus \{z\}$. This also implies that we do not need to care about edges across Z and $X \cup U$, as these edges will be covered by the vertices that are forced into any solution. In the end, we have that the edges whose endpoints are in $X \cup U$ encode the instance ψ.

Consider the vertex $z = \mathsf{pndt}^\star(X_T^\alpha)$. We argue that, in G_{SR}, z is not adjacent to any vertex in $X \cup U$. We prove that, for every vertex u in $X \cup U$, there is a vertex $v \in V(G) \setminus \{u\}$ such that a shortest path between z and v contains u. By Definition 17 and the construction of G_{SR}, this implies that z is not adjacent to u in G_{SR}. First, consider the case where u is in X_T^α. Every vertex in X_T^α is on some shortest path from z to $\mathsf{glb}(X_T^\alpha)$. Hence, z is not adjacent to any vertex in X_T^α in G_{SR}. Now, consider any set $X' \in \{X_T^\delta, X_F^\delta \mid \delta\{\alpha,\beta,\gamma\}\}$ such that $X' \neq X_T^\alpha$. Every vertex in X' is on some shortest path from z to $\mathsf{glb}(X')$. For example, consider the shortest path from z to $\mathsf{glb}(X')$ that contains the vertices $\mathsf{glb}^\star(X_T^\alpha)$, $\mathsf{glb}^\star(X')$, (both of these vertices are part of S_K), and a vertex in X'. Hence, there is no edge incident to z whose other endpoint is in X'. This implies that z is not adjacent to any vertex in X in G_{SR}. The arguments for any set $U' \in \{U^\delta \mid \delta \in \{\alpha,\beta,\gamma\}\}$ are identical. For any $\delta \in \{\alpha,\beta,\gamma\}$, every vertex in U^δ is contained in a shortest path from z to $\mathsf{glb}(U^\delta)$. The

	X_T^α	X_F^α	U_T^α	U_F^α
X_T^α	Ind-Set	Set-Rep	Set-Rep	$\mathsf{glb}(X_T^\alpha)$
X_F^α		Ind-Set	$\mathsf{glb}(X_F^\alpha)$	Set-Rep
U_T^α			Ind-Set	$\mathsf{bit\text{-}rep}(U_T^\alpha)$
U_F^α				Ind-Set
	X_T^δ	X_F^δ	U_T^δ	U_F^δ
X_T^α	$\mathsf{glb}(X_T^\alpha)$	$\mathsf{glb}(X_T^\alpha)$	$\mathsf{glb}(X_T^\alpha)$	$\mathsf{glb}(X_T^\alpha)$
X_F^α		$\mathsf{glb}(X_F^\alpha)$	$\mathsf{glb}(X_F^\alpha)$	$\mathsf{glb}(X_F^\alpha)$
U_T^α			Set-Rep	Set-Rep
U_F^α				Set-Rep

Table 1 Overview of the adjacencies across the partition of $X \cup U$. Here, $\delta \in \{\beta, \gamma\}$. Diagonal entries marked with Ind-Set denote that the set remains as an independent set because of the gadget in Section 7.1.1 and Lemma 23. Set-Rep, Set-Rep, and Set-Rep denote that the corresponding set contains the edges with respect to the connection portal added based on the set representation gadget in Section 7.1.2 and Lemma 24. The other non-empty entries denote that there is no edge across these sets. In particular, for such a non-empty entry, every vertex in the set in the same row (on the far left) lies on a shortest path between the vertex mentioned in the entry and any vertex in the set in the same column (top). For example, every vertex in X_T^α lies on a shortest path between $\mathsf{glb}(X_T^\alpha)$ and any vertex in U_T^α.

other vertices in the path are $\mathsf{glb}^\star(X_T^\delta)$ and $\mathsf{glb}^\star(U^\delta)$ (as both are in the clique S_K). Hence, z is not adjacent to any vertex in U in G_{SR}.

Now, we consider the edges across and within $X \cup U$ in G_{SR}. We consider the following partition of $X \cup U$: X_T^α, X_F^α, X_T^δ, X_F^δ, and U_T^α, U_F^α, U_T^δ, U_F^δ, where $\delta \in \{\beta, \gamma\}$. In Table 1, we describe the edges across and within $X \cup U$ in G_{SR}. In particular, there are two types of non-empty entries. For the first such type, we make reference to a gadget that enforces certain edges to exist and others to not exist in G_{SR}, and the existence and non-existence of these edges is proven by either Lemma 23 or Lemma 24. In the second type of entry, we mention a vertex such that any vertex in the set in the same row (on the far left) lies on a shortest path between the vertex mentioned in the entry and any vertex in the set in the same column (top). In the latter case, this implies that there is no edge in G_{SR} between these two sets. For more details, see the caption of Table 1. For example, consider the entry in the first row and last column. This indicates that there is no edge across X_T^α and U_T^α. Consider the shortest path from u to $\mathsf{glb}(X_T^\alpha)$ that contains the vertices $\mathsf{glb}^\star(U_F^\alpha)$, $\mathsf{glb}^\star(X_T^\alpha)$ (as both these vertices are in the short-cut clique), and any arbitrary vertex x in X_T^α. This implies that there no edge with endpoints u, s in G_{SR}. As these two are arbitrary vertices in the respective sets, our claim holds.

This concludes the proof since G_{SR} is very close to the graph H mentioned in Lemma 21. Indeed, G_{SR} has the properties mentioned in the paragraph following Lemma 21, that is, all the edges in $E(H)$ are present in G_{SR}, all the edges in $E(G_{SR}) \setminus E(H)$ are incident to Z, and $N[z] = Z$. The proof then follows from the remaining arguments in that paragraph. ◀

Proof of Theorem 16. Consider the set Z defined above. As every set mentioned in its definition is of size $\mathcal{O}(\log(n))$, we have $|Z| = \mathcal{O}(\log(n))$. As every vertex in Z is a pendent vertex, $|N(Z)| = \mathcal{O}(\log(n))$. By construction, it is easy to verify that $N(Z)$ is a vertex cover of G. Hence, $\mathsf{vc}(G) = \mathcal{O}(\log(n))$. This implies that if there is an algorithm running in time $2^{2^{o(\mathsf{vc})}} \cdot n^{\mathcal{O}(1)}$, then Exact-3-Partitioned-3-SAT has an algorithm running in time

$2^{o(n)}$, as the reduction takes polynomial time in the size of input. This, however, contradicts Proposition 5. Hence, the first part of the theorem is true. The second part of the theorem follows from similar arguments coupled with the facts that the problem admits a kernel with $2^{\mathcal{O}(\mathtt{vc})}$ vertices and a brute-force algorithm running in time $2^{\mathcal{O}(n)}$. ◀

8 Algorithms

8.1 Dynamic Programming Algorithm for Metric Dimension

The aim of this subsection is to prove the following theorem.

▶ **Theorem 26.** *Metric Dimension admits an algorithm running in time* $2^{\mathtt{diam}^{\mathcal{O}(\mathtt{tw})}} \cdot n^{\mathcal{O}(1)}$.

To this end, we give a dynamic programming algorithm on a tree decomposition for this problem. The algorithm is inspired by the one from [13] for chordal graphs, though there are some non-trivial differences. We will assume that a tree decomposition of the input graph G of width w is given to us. Note that one can compute a tree decomposition of width $w \leq 2\mathtt{tw}(G) + 1$ in time $2^{\mathcal{O}(\mathtt{tw}(G))}n$ [70], and it can be transformed into a *nice tree decomposition* of the same width with $\mathcal{O}(wn)$ bags in time $\mathcal{O}(w^2 n)$ [68].

Overview. As mentioned previously, our dynamic programming is similar to that of [13]. However, in [13], as the diameter of the graph is unbounded, it was crucial to restrict the computations for each step of the dynamic programming to vertices "not too far" from the current bag. This was possible due to the metric properties of chordal graphs. In our case, as we consider the diameter of the graph as a parameter, we do not need such restrictions, which makes the proof a little bit simpler.

We now give an intuitive description of the dynamic programming scheme. At each step of the algorithm, we consider a bounded number of *solution types*, depending on the properties of the solution vertices with respect to the current bag. At a given dynamic programming step, we will assume that the current solution resolves all vertex pairs in G_i. Such a vertex pair may be resolved by a vertex from $G - G_i$, or by a vertex in G_i itself.

Any bag X_i of the tree decomposition whose node i lies on a path between two join nodes in T, forms a separator of G: there are no edges between the vertices of $G_i - X_i$ and $G - G_i$. For a vertex v not in X_i, we consider its distance-vector to the vertices of X_i; the distance-vectors induce an equivalence relation on the vertices of $G - X_i$, whose classes we call X_i*-classes*. Consider the two subgraphs G_i and $G - G_i$. Any two solution vertices x, y from $G - G_i$ that are in the same X_i-class, resolve the exact same pairs of vertices from G_i. Thus, for this purpose, it is irrelevant whether x or y will be in a resolving set, and it is sufficient to know that a vertex of their X_i-class will eventually be chosen. In this way, one can check whether a vertex pair from G_i is resolved by a solution vertex of $G - G_i$.

The same idea is used to "remember" the previously computed solution: it is sufficient to remember the X_i-classes of the vertices in the previously computed resolving set, rather than the vertices themselves.

It is slightly more delicate to make sure that vertex pairs in G_i are resolved in the case where such a pair is resolved by a vertex in G_i. Indeed, this must be ensured, in particular when processing a join node i, for vertex pairs belonging to bags in the two sub-trees corresponding to the children i_1, i_2 of i. Such pairs may be resolved by four types of solution vertices: from $G - G_i$, X_i, $G_{i_1} - X_i$, or $G_{i_2} - X_i$. To ensure this, the dynamic programming scheme makes sure that, at each step, for any possible pair C_1, C_2 of X_i-classes, all vertex pairs $\langle u, v\rangle$ consisting of a vertex u of G_i with class C_1 and a vertex v of $G - G_i$ with class C_2 are resolved. The crucial step here is that when a new vertex v is introduced (i.e., added to a bag X_i to form $X_{i'}$), depending on its X_i-class, it must be made sure that it is resolved from all other vertices depending on their X_i-classes, as described above. To ensure that v is distinguished from all other vertices of G_i, we keep track of vertex pairs of $G_i \times (G - G_i)$ that are already resolved by the partial solution, and enforce that, when processing bag $X_{i'}$, for every vertex x of G_i, the pair $\langle x, v\rangle$ is already resolved. As v belongs to the new bag $X_{i'}$, we know its distances to all resolving vertices (indeed, $X_{i'}$-classes of solution vertices can be computed from their X_i-classes), and thus, the information can be updated accurately.

For a bag X_i and a vertex v not in X_i, the number of possible distance vectors to the vertices of X_i is at most $\mathtt{diam}(G)^{|X_i|}$. Thus, a solution for bag X_i will consist of: (i) the subset of vertices of X_i selected in the solution; (ii) a subset of the $\mathtt{diam}(G)^{|X_i|}$ possible vectors to denote the X_i-classes from which the currently computed solution (for G_i) contains at least one vertex in the resolving set; (iii) a subset of the $\mathtt{diam}(G)^{|X_i|}$ possible vectors denoting the X_i-classes from which the future solution needs at least one vertex of $G - G_i$ in the resolving set; (iv) a subset of the $\mathtt{diam}(G)^{|X_i|} \times \mathtt{diam}(G)^{|X_i|}$ possible pairs of vectors representing the X_i-classes of the pairs of vertices in $G_i \times (G - G_i)$ that are already resolved by the partial solution.

Formal description. We mostly follow the notations used in [13]. Before presenting the dynamic program, we first introduce some useful definitions and lemmas.

▶ **Definition 27.** *Given a vector* $\mathbf{r}$*, we refer to the i-th coordinate of* $\mathbf{r}$ *as* $\mathbf{r}_i$*.*

- *Let* $\mathbf{r}$ *be a vector of size* k *and let* m *be an integer. The vector* $\mathbf{t} = \mathbf{r}|m$ *is the vector of size* $k+1$ *such that* $\mathbf{t}_{k+1} = m$ *and, for all* $1 \leq i \leq k$, $\mathbf{t}_i = \mathbf{r}_i$ *.*
- *Let* $\mathbf{r}$ *be a vector of size* k*. The vector* $\mathbf{r}^-$ *is the vector of size* $k-1$ *such that, for all* $1 \leq i \leq k-1$, $\mathbf{r}^-_i = \mathbf{r}_i$*.*

▶ **Definition 28.** *Let* G *be a graph and let* $X = \{v_1, \ldots, v_k\}$ *be a subset of vertices of* G*. For a vertex* x *of* G*, the* distance vector $\mathbf{d_X}(x)$ *of* x *to* X *is the vector of size* k *such that, for all* $1 \leq j \leq k$, $\mathbf{d_X}(x)_j = d(x, v_j)$*. For a set* $S \subseteq V(G)$*, we let* $\mathbf{d_X}(S) = \{\mathbf{d_X}(s) \mid s \in S\}$*.*

▶ **Definition 29.** *Let* $\mathbf{r_1}, \mathbf{r_2}$*, and* $\mathbf{r_3}$ *be three vectors of size* k*. We say that* $\mathbf{r_3}$ resolves *the pair* $\langle \mathbf{r_1}, \mathbf{r_2}\rangle$ *if*

$$\min_{1\leq i\leq k} (\mathbf{r_1} + \mathbf{r_3})_i \neq \min_{1\leq i\leq k} (\mathbf{r_2} + \mathbf{r_3})_i.$$

▶ **Lemma 30.** *Let* X *be a separator of a graph* G*, and let* G_1 *be a connected component of* $G - X$*. Let* $\langle x, y\rangle$ *be a pair of vertices of* $G - G_1$*, and let* $\mathbf{r}$ *be a vector of size* $|X|$*. If* $\mathbf{r}$ *resolves the pair* $\langle \mathbf{d_X}(x), \mathbf{d_X}(y)\rangle$*, then any vertex* $s \in V(G_1)$ *such that* $\mathbf{d_X}(s) = \mathbf{r}$ *resolves the pair* $\langle x, y\rangle$*.*

Proof. To see this, it suffices to note that since X separates s from x (y, resp.), $d(s,x) = \min_{1\leq j\leq |X|}(\mathbf{d_X}(s) + \mathbf{d_X}(x))_j$ ($d(s,y) = \min_{1\leq j\leq |X|}(\mathbf{d_X}(s) + \mathbf{d_X}(y))_j$, resp.); and since $\mathbf{r}$ resolves the pair $\langle \mathbf{d_X}(x), \mathbf{d_X}(y)\rangle$, $d(s,x) \neq d(s,y)$. ◀

▶ **Definition 31.** *Let* X *be a separator of a graph* G*, and let* G_1, G_2 *be two (not necessarily distinct) connected components of* $G - X$*. Let* $x \in V(G_1) \cup X$ *and* $y \in V(G_2) \cup X$*. If a vector* $\mathbf{r}$ *resolves the pair* $\langle \mathbf{d_X}(x), \mathbf{d_X}(y)\rangle$*, then we say that* $\mathbf{r}$ resolves *the pair* $\langle x, y\rangle$*. More generally, given a set* M *of vectors, we say that the pair* $\langle x, y\rangle$ is resolved *by* M *if there exists* $\mathbf{r} \in M$ *that resolves the pair* $\langle x, y\rangle$*.*

We now define the generalized problem solved at each step of the dynamic programming algorithm, called Extended Metric Dimension (EMD for short), whose instances are defined as follows.

▶ **Definition 32.** *Let G be a graph and let $(T, \{X_i : i \in V(T)\})$ be a tree decomposition of G. For a node i of T, an* instance of EMD *for i is a 5-tuple $I = (X_i, S_I, D_{int}(I), D_{ext}(I), D_{pair}(I))$ composed of the bag X_i of i, a subset S_I of X_i, and three sets of vectors satisfying the following.*

- $D_{int}(I) \subseteq [\mathtt{diam}(G)]^{|X_i|}$ *and* $D_{ext}(I) \subseteq [\mathtt{diam}(G)]^{|X_i|}$.
- $D_{ext}(I) \neq \emptyset$ *or* $S_I \neq \emptyset$.
- $D_{pair}(I) \subseteq [\mathtt{diam}(G)]^{|X_i|} \times [\mathtt{diam}(G)]^{|X_i|}$.
- *For each pair of vectors $\langle \mathbf{r_1}, \mathbf{r_2} \rangle \in D_{pair}(I)$, there exist two vertices $x \in V(G_i)$ and $y \notin V(G_i)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r_1}$ and $\mathbf{d_{X_i}}(y) = \mathbf{r_2}$.*
- *For each vector $\mathbf{r}$ of $D_{ext}(I)$, there exists $x \notin V(G_i)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$.*

▶ **Definition 33.** *A set $S \subseteq G_i$ is a solution for an instance I of* EMD *if the following hold.*

- ***(S1)*** *Every pair of vertices of G_i is either resolved by a vertex in S or resolved by a vector of $D_{ext}(I)$.*
- ***(S2)*** *For each vector $\mathbf{r} \in D_{int}(I)$, there exists a vertex $s \in S$ such that $\mathbf{d_{X_i}}(s) = \mathbf{r}$.*
- ***(S3)*** *For each pair of vectors $\langle \mathbf{r_1}, \mathbf{r_2} \rangle \in D_{pair}(I)$, any vertex $x \in V(G_i)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r_1}$, and any vertex $y \notin V(G_i)$ such that $\mathbf{d_{X_i}}(y) = \mathbf{r_2}$, the pair $\langle x, y \rangle$ is resolved by S.*
- ***(S4)*** $S \cap X_i = S_I$.

In the remainder of this section, for brevity, we will refer to an instance of the EMD problem only by an instance.

▶ **Definition 34.** *Let I be an instance. We denote by* $\dim(I)$ *the minimum size of a set $S \subseteq V(G_i)$ which is a solution for I. If no such set exists, then we set* $\dim(I) = +\infty$. *We refer to this value as the* extended metric dimension *of I.*

In the following, we fix a graph G and a nice tree decomposition $(T, \{X_i : i \in V(T)\})$ of G. Given a node i of T and an instance I for i, we show how to compute $\dim(I)$. The proof is divided according to the type of the node i.

Leaf node. Computing $\dim(I)$ when I is an instance for a leaf node can be done with the following lemma.

▶ **Lemma 35.** *Let I be an instance for a leaf node i and let v be the only vertex in X_i. Then,*

$$\dim(I) = \begin{cases} 0 & \textit{if } S_I = \emptyset, D_{int}(I) = \emptyset, \textit{ and } D_{pair}(I) = \emptyset \\ 1 & \textit{if } S_I = \{v\} \textit{ and } D_{int}(I) \subseteq \{(0)\} \\ +\infty & \textit{otherwise} \end{cases}$$

Proof. Suppose first that $S_I = \emptyset$. Then, the empty set is the only possible solution for I; and the empty set is a solution for I only if $D_{int}(I) = \emptyset$ and $D_{pair}(I) = \emptyset$. Suppose next that $S_I = \{v\}$. Then, the set $S = \{v\}$ is the only possible solution for I; and this set is a solution for I only if $D_{int}(I) = \emptyset$ or $D_{int}(I)$ contains only the vector $\mathbf{d_{X_i}}(v) = (0)$. ◀

In the remainder of this section, we handle the three other types of nodes. For each type of node, we proceed as follows: we first define a notion of compatibility on the instances for the child/children of i and show how to compute the extended metric dimension of I from the extended metric dimension of compatible instances for the child/children of i.

Join node. Let I be an instance for a join node i, and let i_1 and i_2 be the two children of i. Given a pair of instances $\langle I_1, I_2 \rangle$ for $\langle i_1, i_2 \rangle$, we say that a pair $\langle \mathbf{r}, \mathbf{t} \rangle \in [\mathtt{diam}(G)]^{|X_i|} \times$

$[\mathtt{diam}(G)]^{|X_i|}$ is *2-compatible* if there exist $x \in V(G_{i_1}), y \in V(G - G_i)$, and $\mathbf{u} \in D_{int}(I_2)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, $\mathbf{d_{X_i}}(y) = \mathbf{t}$, and $\mathbf{u}$ resolves the pair $\langle \mathbf{r}, \mathbf{t} \rangle$. Symmetrically, we call a pair $\langle \mathbf{r}, \mathbf{t} \rangle \in [\mathtt{diam}(G)]^{|X_i|} \times [\mathtt{diam}(G)]^{|X_i|}$ *1-compatible* if there exist $x \in V(G_{i_2}), y \in V(G - G_i)$, and $\mathbf{u} \in D_{int}(I_1)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, $\mathbf{d_{X_i}}(y) = \mathbf{t}$, and $\mathbf{u}$ resolves the pair $\langle \mathbf{r}, \mathbf{t} \rangle$.

▶ **Definition 36.** *A pair of instances $\langle I_1, I_2 \rangle$ for $\langle i_1, i_2 \rangle$ is* compatible *with I if the following hold.*

- ***(J1)*** $S_{I_1} = S_{I_2} = S_I$.
- ***(J2)*** $D_{ext}(I_1) \subseteq D_{ext}(I) \cup D_{int}(I_2)$ *and* $D_{ext}(I_2) \subseteq D_{ext}(I) \cup D_{int}(I_1)$.
- ***(J3)*** $D_{int}(I) \subseteq D_{int}(I_1) \cup D_{int}(I_2)$.
- ***(J4)*** *Let* $C_1 = \{\langle \mathbf{r}, \mathbf{t} \rangle \in [\mathtt{diam}(G)]^{|X_i|} \times [\mathtt{diam}(G)]^{|X_i|} \mid \nexists x \in V(G_{i_1})$ *s.t.* $\mathbf{d_{X_{i_1}}}(x) = \mathbf{r}\}$ *and* $C_2 = \{\langle \mathbf{r}, \mathbf{t} \rangle \in [\mathtt{diam}(G)]^{|X_i|} \times [\mathtt{diam}(G)]^{|X_i|} \mid \nexists y \in V(G_{i_2})$ *s.t.* $\mathbf{d_{X_{i_2}}}(y) = \mathbf{r}\}$. *Further, let* $D_1 = \{\langle \mathbf{r}, \mathbf{t} \rangle \in [\mathtt{diam}(G)]^{|X_i|} \times [\mathtt{diam}(G)]^{|X_i|} \mid \langle \mathbf{r}, \mathbf{t} \rangle$ *is 2-compatible*$\}$ *and* $D_2 = \{\langle \mathbf{r}, \mathbf{t} \rangle \in [\mathtt{diam}(G)]^{|X_i|} \times [\mathtt{diam}(G)]^{|X_i|} \mid \langle \mathbf{r}, \mathbf{t} \rangle$ *is 1-compatible*$\}$. *Then,* $D_{pair}(I) \subseteq (C_1 \cup D_1 \cup D_{pair}(I_1)) \cap (C_2 \cup D_2 \cup D_{pair}(I_2))$.
- ***(J5)*** *For all* $\mathbf{r_1}, \mathbf{r_2} \in [\mathtt{diam}(G)]^{|X_i|}$ *for which there exist* $x \in V(G_{i_1})$ *and* $y \in V(G_{i_2})$ *such that* $\mathbf{d_{X_{i_1}}}(x) = \mathbf{r_1}$ *and* $\mathbf{d_{X_{i_2}}}(y) = \mathbf{r_2}$, *one of the following holds:*
  - $\langle \mathbf{r_1}, \mathbf{r_2} \rangle \in D_{pair}(I_1)$,
  - $\langle \mathbf{r_2}, \mathbf{r_1} \rangle \in D_{pair}(I_2)$, *or*
  - *there exists* $\mathbf{t} \in D_{ext}(I)$ *such that* $\mathbf{t}$ *resolves the pair* $\langle \mathbf{r_1}, \mathbf{r_2} \rangle$.

Let $\mathcal{F}_J(I)$ be the set of pairs of instances compatible with I. We aim to prove the following.

▶ **Lemma 37.** *Let I be an instance for a join node i. Then,*

$$\dim(I) = \min_{(I_1,I_2) \in \mathcal{F}_J(I)} (\dim(I_1) + \dim(I_2) - |S_I|).$$

To prove Lemma 37, we prove the following two lemmas.

▶ **Lemma 38.** *Let $\langle I_1, I_2 \rangle$ be a pair of instances for $\langle i_1, i_2 \rangle$ compatible with I such that* $\dim(I_1)$ *and* $\dim(I_2)$ *have finite values. Let S_1 be a minimum-size solution for I_1 and S_2 a minimum-size solution for I_2. Then, $S = S_1 \cup S_2$ is a solution for I. In particular,*

$$\dim(I) \le \min_{(I_1,I_2) \in \mathcal{F}_J(I)} (\dim(I_1) + \dim(I_2) - |S_I|).$$

Proof. Let us show that every condition of Definition 33 is satisfied.

(S1) Let $\langle x, y \rangle$ be a pair of vertices of G_i. Assume first that $x, y \in V(G_{i_1})$. Then, since S_1 is a solution for I_1, either S_1 resolves the pair $\langle x, y \rangle$, in which case we are done; or the pair $\langle x, y \rangle$ is resolved by a vector $\mathbf{t} \in D_{ext}(I_1)$. In the latter case, by compatibility, either $\mathbf{t} \in D_{ext}(I)$, in which case the pair $\langle x, y \rangle$ is still resolved by $\mathbf{t} \in D_{ext}(I)$, or $\mathbf{t} \in D_{int}(I_2)$; but then, there exists $s \in S_2$ such that $\mathbf{d_{X_{i_2}}}(s) = \mathbf{t}$, and so, $s \in S$ resolves the pair $\langle x, y \rangle$. The case where $x, y \in V(G_{i_2})$ is handled symmetrically.

Assume therefore that $x \in V(G_{i_1})$ and $y \in V(G_{i_2})$. Then, by compatibility, one of the following holds:

1. $\langle \mathbf{d_{X_{i_1}}}(x), \mathbf{d_{X_{i_2}}}(y) \rangle \in D_{pair}(I_1)$,
2. $\langle \mathbf{d_{X_{i_2}}}(y), \mathbf{d_{X_{i_1}}}(x) \rangle \in D_{pair}(I_2)$, or
3. there exists $\mathbf{t} \in D_{ext}(I)$ such that $\mathbf{t}$ resolves the pair $\langle \mathbf{d_{X_{i_1}}}(x), \mathbf{d_{X_{i_2}}}(y) \rangle$.

Suppose that item 3. does not hold (we are done otherwise). If item 1. holds, then, since S_1 is a solution for I_1, the pair $\langle x, y\rangle$ is resolved by S_1; and we conclude symmetrically if item 2. holds.

(S2) Consider $\mathbf{r} \in D_{int}(I)$. Then, by compatibility, $\mathbf{r} \in D_{int}(I_1)$ or $\mathbf{r} \in D_{int}(I_2)$. If the former holds, then since S_1 is a solution for I_1, there exists $s \in S_1$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s) = \mathbf{r}$; but then, $s \in S$ with $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(s) = \mathbf{r}$. We conclude similarly if $\mathbf{r} \in D_{int}(I_2)$.

(S3) Consider $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$ and let $x \in V(G_i)$, $y \in V(G - G_i)$ be such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(x) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(y) = \mathbf{t}$. Assume, without loss of generality, that $x \in V(G_{i_1})$. By compatibility, $\langle \mathbf{r}, \mathbf{t}\rangle \in (C_1 \cup D_1 \cup D_{pair}(I_1)) \cap (C_2 \cup D_2 \cup D_{pair}(I_2))$; in particular, $\langle \mathbf{r}, \mathbf{t}\rangle \in C_1 \cup D_1 \cup D_{pair}(I_1)$. Note that $\langle \mathbf{r}, \mathbf{t}\rangle \notin C_1$ since $x \in V(G_{i_1})$ and $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(x) = \mathbf{r}$. Now, suppose that $\langle \mathbf{r}, \mathbf{t}\rangle \in D_1$. Then, there exists $\mathbf{u} \in D_{int}(I_2)$ such that $\mathbf{u}$ resolves the pair $\langle \mathbf{r}, \mathbf{t}\rangle$; and since S_2 is a solution for I_2, there exists $s \in S_2$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_2}}(s) = \mathbf{u}$, and so, $s \in S$ resolves the pair $\langle x, y\rangle$. Finally, if $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I_1)$, then S_1 resolves $\langle x, y\rangle$ as it is a solution for I_1.

(S4) By compatibility, $S_{I_1} = S_{I_2} = S_I$, and thus, $S \cap X_i = S \cap X_{i_1} = S_{I_1} = S_I$.

It now follows from the above that $\dim(I) \le \dim(I_1) + \dim(I_2) - |S_I|$, and since this holds true for any $\langle I_1, I_2\rangle \in \mathcal{F}_J(I)$, the lemma follows. ◀

▶ **Lemma 39.** *Let I be an instance for a join node i. Then,*

$$\dim(I) \ge \min_{(I_1,I_2)\in\mathcal{F}_J(I)} (\dim(I_1) + \dim(I_2) - |S_I|).$$

Proof. If $\dim(I) = +\infty$, then the inequality readily holds. Thus, assume that $\dim(I) < +\infty$, and let S be a minimum-size solution for I. For $j \in \{1, 2\}$, let $S_j = S \cap V(G_{i_j})$. Now, let I_1 and I_2 be the two instances for i_1 and i_2, respectively, defined as follows.

- $S_{I_1} = S_{I_2} = S_I$.
- $D_{int}(I_1) = \mathbf{d}_{\mathbf{X}_\mathbf{i}}(S_1)$ and $D_{int}(I_2) = \mathbf{d}_{\mathbf{X}_\mathbf{i}}(S_2)$.
- $D_{ext}(I_1) = D_{ext}(I) \cup D_{int}(I_2)$ and $D_{ext}(I_2) = D_{ext}(I) \cup D_{int}(I_1)$.
- We construct $D_{pair}(I_1)$ as follows ($D_{pair}(I_2)$ is constructed symmetrically). For every $\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_{i_1}|})^2$, let $R_{\langle \mathbf{r},\mathbf{t}\rangle} = \{\langle x, y\rangle \in V(G_{i_1}) \times V(G - G_{i_1}) \mid \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(y) = \mathbf{t}\}$. If, for every pair $\langle x, y\rangle \in R_{\langle \mathbf{r},\mathbf{t}\rangle}$, S_1 resolves $\langle x, y\rangle$, then we add $\langle \mathbf{r}, \mathbf{t}\rangle$ to $D_{pair}(I_1)$.

Let us show that $\langle I_1, I_2\rangle$ is compatible with I and that, for $j \in \{1, 2\}$, S_j is a solution for I_j.

▷ Claim 40. The constructed pair of instances $\langle I_1, I_2\rangle$ for $\langle i_1, i_2\rangle$ is compatible with I.

Proof. It is clear that conditions **(J1)**, **(J2)**, and **(J3)** of Definition 36 hold; let us show that the remaining conditions hold as well.

(J4) Consider a pair $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$. Let us show that $\langle \mathbf{r}, \mathbf{t}\rangle \in C_1 \cup D_1 \cup D_{pair}(I_1)$ (showing that $\langle \mathbf{r}, \mathbf{t}\rangle \in C_2 \cup D_2 \cup D_{pair}(I_2)$ can be done symmetrically).

If there exists no vertex $x \in V(G_{i_1})$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}$, then $\langle \mathbf{r}, \mathbf{t}\rangle \in C_1$. Otherwise, let $x \in V(G_{i_1})$ be a vertex such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}$, and let $y \in V(G - G_i)$ be a vertex such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(y) = \mathbf{t}$ (note that such a vertex y exists since $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$). Then, since S is a solution for I, $\langle x, y\rangle$ is resolved by S. Now, if there exists a vertex $s \in S \cap V(G_{i_2})$ such that s resolves $\langle x, y\rangle$, then s resolves every pair $\langle u, v\rangle \in V(G_{i_1}) \times V(G - G_i)$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(u) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(v) = \mathbf{t}$; but then, $\langle \mathbf{r}, \mathbf{t}\rangle \in D_1$. Thus, assume that no vertex in

$S \cap V(G_{i_2})$ resolves $\langle x, y\rangle$. Then, there exists a vertex $s \in S \cap V(G_{i_1})$ that resolves the pair $\langle x, y\rangle$; and since this holds for every pair $\langle u, v\rangle \in V(G_{i_1}) \times V(G - G_i)$ (and, a fortiori, for every pair $\langle u, v\rangle \in V(G_{i_1}) \times V(G - G_{i_1})$) such that $\mathbf{d_{X_{i_1}}}(u) = \mathbf{r}$ and $\mathbf{d_{X_{i_1}}}(v) = \mathbf{t}$, we conclude that $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I_1)$.

(J5) Let $\mathbf{r_1}, \mathbf{r_2} \in [\mathtt{diam}(G)]^{|X_i|}$ be two vectors for which there exist $x \in V(G_{i_1})$ and $y \in V(G_{i_2})$ such that $\mathbf{d_{X_{i_1}}}(x) = \mathbf{r_1}$ and $\mathbf{d_{X_{i_2}}}(y) = \mathbf{r_2}$. Then, since S is a solution for I, either the pair $\langle x, y\rangle$ is resolved by a vector in $D_{ext}(I)$, in which case condition (J5) holds; or there exists a vertex in S resolving $\langle x, y\rangle$. Let us show that, in the latter case, $\langle \mathbf{r_1}, \mathbf{r_2}\rangle \in D_{pair}(I_1)$ or $\langle \mathbf{r_2}, \mathbf{r_1}\rangle \in D_{pair}(I_2)$. Suppose toward a contradiction that this does not hold. Then, there exist $\langle x_1, y_1\rangle \in V(G_{i_1}) \times V(G_{i_2})$ and $\langle x_2, y_2\rangle \in V(G_{i_2}) \times V(G_{i_1})$ such that $\mathbf{d_{X_{i_1}}}(x_1) = \mathbf{d_{X_{i_2}}}(y_2) = \mathbf{r_1}$, $\mathbf{d_{X_{i_1}}}(y_1) = \mathbf{d_{X_{i_2}}}(x_2) = \mathbf{r_2}$, S_1 does not resolve $\langle x_1, y_1\rangle$, and S_2 does not resolve $\langle x_2, y_2\rangle$. Now, since S is a solution for I, there exists $s \in S$ such that s resolves the pair $\langle x_1, x_2\rangle$; but then, either $s \in S_1$, in which case s resolves the pair $\langle x_1, y_1\rangle$, or $s \in S_2$, in which case s resolves the pair $\langle x_2, y_2\rangle$, a contradiction in both cases. ◁

▷ Claim 41. For every $j \in \{1, 2\}$, S_j is a solution for I_j.

Proof. We only prove that S_1 is a solution for I_1 as the other case is symmetric. To this end, let us show that every condition of Definition 33 is satisfied.

(S1) Consider two vertices $x, y \in V(G_{i_1})$. Since S is a solution for I, the pair $\langle x, y\rangle$ is either resolved by a vector in $D_{ext}(I)$, in which case we are done as $D_{ext}(I) \subseteq D_{ext}(I_1)$ by construction; or resolved by a vertex $s \in S$. Now, if $s \in V(G_{i_1})$, then s is a vertex of S_1 resolving $\langle x, y\rangle$. Otherwise, $s \in V(G_{i_2})$ and by construction of I_1, there exists a vector $\mathbf{r} \in D_{ext}(I_1)$ such that $\mathbf{d_{X_i}}(s) = \mathbf{r}$, and so, $\mathbf{r}$ resolves the pair $\langle x, y\rangle$.

(S2) Readily follows from the fact that $D_{int}(I_1) = \mathbf{d_{X_i}}(S_1)$.

(S3) By construction, for every $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I_1)$, any $x \in V(G_{i_1})$ such that $\mathbf{d_{X_{i_1}}}(x) = \mathbf{r}$, and any $y \notin V(G_{i_1})$ such that $\mathbf{d_{X_{i_1}}}(y) = \mathbf{t}$, there exists $s \in S_1$ such that s resolves the pair $\langle x, y\rangle$.

(S4) By construction, $S_{I_1} = S_I$, and thus, $S \cap X_{i_1} = S \cap X_i = S_I = S_{I_1}$. ◁

To conclude, since the sets S_1 and S_2 are solutions for I_1 and I_2, respectively, we have that $\dim(I_1) \leq |S_1|$ and $\dim(I_2) \leq |S_2|$. Now, $|S| = |S_1| + |S_2| - |S_I|$, and so, $|S| = \dim(I) \geq \dim(I_1) + \dim(I_2) - |S_I| \geq \min_{\langle J_1, J_2\rangle \in \mathcal{F}_J(I)}(\dim(J_1) + \dim(J_2) - |S_I|)$. ◀

Introduce node. Let I be an instance for an introduce node i with child i_1, and let $v \in V(G)$ be such that $X_i = \{v\} \cup X_{i_1}$. Further, let $X_i = \{v_1, \ldots, v_k\}$, where $v = v_k$.

▶ **Definition 42.** *An instance I_1 for i_1 is compatible with I of* type 1 *if the following hold.*

- **(I1)** $S_I = S_{I_1}$.
- **(I2)** *For all $\mathbf{r} \in D_{ext}(I_1)$, there exists $\mathbf{t} \in D_{ext}(I)$ such that $\mathbf{t}^- = \mathbf{r}$.*
- **(I3)** *For all $\mathbf{r} \in D_{int}(I)$, $\mathbf{r}_k = \min_{1 \leq \ell \leq k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_\ell$ and $\mathbf{r}^- \in D_{int}(I_1)$.*
- **(I4)** *Let $P_1 = \{\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_i|})^2 \mid \mathbf{r}_k \geq 1$ and $\langle \mathbf{r}^-, \mathbf{t}^-\rangle \in D_{pair}(I_1)\}$ and $C_1 = \{\langle \mathbf{d_{X_i}}(v), \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_i|})^2 \mid \exists \mathbf{u} \in D_{int}(I_1), \mathbf{u}$ resolves $\langle \mathbf{d_{X_i}}(v)^-, \mathbf{t}^-\rangle\}$. Then, $D_{pair}(I) \subseteq P_1 \cup C_1$.*
- **(I5)** *For all $\mathbf{r} \in [\mathtt{diam}(G)]^{|X_i|}$ such that*

- *there exists* $x \in V(G_{i_1})$ *with* $\mathbf{d_{X_i}}(x) = \mathbf{r}$*, and*
- *no vector in* $D_{ext}(I)$ *resolves the pair* $\langle x, v\rangle$*,*

$\langle \mathbf{r}^-, \mathbf{d_{X_{i_1}}}(v)\rangle \in D_{pair}(I_1)$*.*

An instance I_1 *for* i_1 *is compatible with* I *of* type 2 *if the following hold.*

- **(I'1)** $S_I = S_{I_1} \cup \{v\}$*.*
- **(I'2)** $\mathbf{d_{X_{i_1}}}(v) \in D_{ext}(I_1)$ *and, for all* $\mathbf{r} \in D_{ext}(I_1) \setminus \{\mathbf{d_{X_{i_1}}}(v)\}$*, there exists* $\mathbf{t} \in D_{ext}(I)$ *such that* $\mathbf{t}^- = \mathbf{r}$*.*
- **(I'3)** *For all* $\mathbf{r} \in D_{int}(I) \setminus \{\mathbf{d_{X_i}}(v)\}$*,* $\mathbf{r}_k = \min_{1\le \ell \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_\ell$ *and* $\mathbf{r}^- \in D_{int}(I_1)$*.*
- **(I'4)** *Let* $P_2 = \{\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_i|})^2 \mid \mathbf{r}_k \ge 1$ *and* $\langle \mathbf{r}^-, \mathbf{t}^-\rangle \in D_{pair}(I_1)\}$ *and* $C_2 = \{\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_i|})^2 \mid \mathbf{r}_k \ne \mathbf{t}_k)\}$*. Then,* $D_{pair}(I) \subseteq P_2 \cup C_2$*.*

We denote by $\mathcal{F}_1(I)$ the set of instances for i_1 compatible with I of type 1, and by $\mathcal{F}_2(I)$ the set of instances for i_1 compatible with I of type 2. We aim to prove the following.

▶ **Lemma 43.** *Let* I *be an instance for an introduce node* i*. Then,*

$$\dim(I) = \min \{ \min_{I_1 \in \mathcal{F}_1(I)} \{\dim(I_1)\}, \min_{I_2 \in \mathcal{F}_2(I)} \{\dim(I_2) + 1\}\}.$$

To prove Lemma 43, we prove the following lemmas.

▶ **Lemma 44.** *Let* I_1 *be an instance for* i_1 *compatible with* I *of type 1, and let* S *be a minimum-size solution for* I_1*. Then,* S *is a solution for* I*.*

Proof. Let us prove that every condition of Definition 33 is satisfied.

(S1) Let $\langle x, y\rangle$ be a pair of vertices of G_i. Assume first that $x \ne v$ and $y \ne v$, and suppose that S does not resolve $\langle x, y\rangle$ (we are done otherwise). Then, since S is a solution for I_1 and I_1 is compatible with I, there exists $\mathbf{r} \in D_{ext}(I)$ such that $\mathbf{r}^- \in D_{ext}(I_1)$ and $\mathbf{r}^-$ resolves the pair $\langle x, y\rangle$; but then, $\mathbf{r}$ resolves $\langle x, y\rangle$ as well. Assume next that $x = v$ and suppose that no vector in $D_{ext}(I)$ resolves the pair $\langle x, y\rangle$ (we are done otherwise). Then, since I_1 is compatible with I of type 1, $\langle \mathbf{d_{X_{i_1}}}(y), \mathbf{d_{X_{i_1}}}(x)\rangle \in D_{pair}(I_1)$, and so, S resolves the pair $\langle x, y\rangle$ as it is a solution for I_1.

(S2) Consider $\mathbf{r} \in D_{int}(I)$. Since I_1 is compatible with I, there exists $\mathbf{t} \in D_{int}(I_1)$ such that $\mathbf{r} = \mathbf{t}|\min_{1\le\ell\le k-1}(\mathbf{t} + \mathbf{d_{X_{i_1}}}(v))_\ell$. Now, since S is a solution for I_1, there exists $s \in S$ such that $\mathbf{d_{X_{i_1}}}(s) = \mathbf{t}$; but then, $\mathbf{d_{X_i}}(s) = \mathbf{r}$ as $d(s, v) = \min_{1\le\ell\le k-1}(\mathbf{t} + \mathbf{d_{X_{i_1}}}(v))_\ell$ (indeed, X_{i_1} separates v from s).

(S3) Consider $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$. Let $x \in V(G_i)$ be such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, and let $y \notin V(G_i)$ be such that $\mathbf{d_{X_i}}(y) = \mathbf{t}$. Then, since I_1 is compatible with I of type 1, $\langle \mathbf{r}, \mathbf{t}\rangle \in P_1$ or $\langle \mathbf{r}, \mathbf{t}\rangle \in C_1$. Now, if the former holds, then, $x \ne v$ and $\langle \mathbf{r}^-, \mathbf{t}^-\rangle \in D_{pair}(I_1)$, and so, S resolves the pair $\langle x, y\rangle$ as it is a solution for I_1. Suppose therefore that the latter holds. Then, $x = v$ and there exists $\mathbf{u} \in D_{int}(I_1)$ such that $\mathbf{u}$ resolves the pair $\langle x, y\rangle$; but S is a solution for I_1, and thus, there exists $s \in S$ such that $\mathbf{d_{X_{i_1}}}(s) = \mathbf{u}$.

(S4) By compatibility of type 1, $S_I = S_{I_1}$, and thus, $S \cap X_i = S \cap (X_{i_1} \cup \{v\}) = S_{I_1} = S_I$. ◀

▶ **Lemma 45.** *Let* I_1 *be an instance for* i_1 *compatible with* I *of type 2, and let* S *be a minimum-size solution for* I_1*. Then,* $S \cup \{v\}$ *is a solution for* I*.*

Proof. Let us prove that the conditions of Definition 33 are satisfied. In the following, we let $S' = S \cup \{v\}$.

(S1) Let $\langle x, y\rangle$ be a pair of vertices of G_i such that $x \neq v$ and $y \neq v$ (it is otherwise clear that the pair is resolved by $v \in S'$). Suppose that S does not resolve the pair $\langle x, y\rangle$ (we are done otherwise). Then, since S is a solution for I_1, $\langle x, y\rangle$ is resolved by a vector $\mathbf{r} \in D_{ext}(I_1)$. Now, I_1 is compatible with I, and thus, there exists $\mathbf{t} \in D_{ext}(I)$ such that $\mathbf{t}^- = \mathbf{r}$; but then, $\mathbf{t}$ resolves the pair $\langle x, y\rangle$ as well.

(S2) Consider $\mathbf{r} \in D_{int}(I)$ and suppose that $\mathbf{r} \neq \mathbf{d}_{\mathbf{X_i}}(v)$ (otherwise $v \in S'$ has $\mathbf{r}$ as its distance vector to X_i). Then, since I_1 is compatible with I, there exists $\mathbf{t} \in D_{int}(I_1)$ such that $\mathbf{r} = \mathbf{t}| \min_{1\leq \ell \leq k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_\ell$. Now, S is a solution for I_1, and thus, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s) = \mathbf{t}$; but then, $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{r}$ as $d(s, v) = \min_{1\leq \ell \leq k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_\ell$ (indeed, X_{i_1} separates v from s).

(S3) Consider $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$. Let $x \in V(G_i)$ be such that $\mathbf{d}_{\mathbf{X_i}}(x) = \mathbf{r}$, and let $y \notin V(G_i)$ be such that $\mathbf{d}_{\mathbf{X_i}}(y) = \mathbf{t}$. Suppose that v does not resolve the pair $\langle x, y\rangle$ (we are done otherwise). Then, $\mathbf{r}_k = d(x, v) = d(y, v) = \mathbf{t}_k \geq 1$, which implies that $\langle \mathbf{r}^-, \mathbf{t}^-\rangle \in D_{pair}(I_1)$ as I_1 is compatible with I of type 2. But S is a solution for I_1, and so, S (and, a fortiori, S') resolves the pair $\langle x, y\rangle$.

(S4) By compatibility of type 2, $S_I = S_{I_1} \cup \{v\}$, and thus, $S \cap X_i = S \cap (X_{i_1} \cup \{v\}) = S_{I_1} \cup \{v\} = S_I$. ◀

As a consequence of Lemmas 44 and 45, the following holds.

▶ **Lemma 46.** *Let I be an instance for an introduce node i. Then,*

$$\dim(I) \leq \min\{\min_{I_1\in\mathcal{F}_1(I)}\{\dim(I_1)\}, \min_{I_2\in\mathcal{F}_2(I)}\{\dim(I_2) + 1\}\}.$$

▶ **Lemma 47.** *Let S be a minimum-size solution for I such that $v \notin S$. Then, there exists $I_1 \in \mathcal{F}_1(I)$ such that S is a solution for I_1.*

Proof. Let I_1 be the instance for i_1 defined as follows.

- $S_{I_1} = S_I$, $D_{ext}(I_1) = \{\mathbf{r}^- \mid \mathbf{r} \in D_{ext}(I)\}$, and $D_{int}(I_1) = \mathbf{d}_{\mathbf{X_{i_1}}}(S)$.
- For any $\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_{i_1}|})^2$, let $R_{\langle \mathbf{r},\mathbf{t}\rangle} = \{\langle x, y\rangle \in V(G_{i_1}) \times V(G - G_{i_1}) \mid \mathbf{d}_{\mathbf{X_{i_1}}}(x) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X_{i_1}}}(y) = \mathbf{t}\}$. If S resolves every pair in $R_{\langle \mathbf{r},\mathbf{t}\rangle}$, then we add $\langle \mathbf{r}, \mathbf{t}\rangle$ to $D_{pair}(I_1)$.

Let us prove that $I_1 \in \mathcal{F}_1(I)$ and that S is a solution for I_1.

▷ Claim 48. The constructed instance I_1 is compatible with I of type 1.

Proof. It is clear that conditions **(I1)** and **(I2)** of Definition 42 hold; let us show that the remaining conditions hold as well.

(I3) Since S is a solution for I, for every $\mathbf{r} \in D_{int}(I)$, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{r}$; but then, $\mathbf{r}^- = \mathbf{d}_{\mathbf{X_{i_1}}}(s)$ and $\mathbf{r}_k = d(s, v) = \min_{1\leq \ell\leq k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_\ell$ as $v \notin S$ and X_{i_1} separates s from v.

(I4) Consider $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$ and assume first that $\mathbf{r} \neq \mathbf{d}_{\mathbf{X_i}}(v)$. Then, for any $x \in V(G_i)$ such that $\mathbf{d}_{\mathbf{X_i}}(x) = \mathbf{r}$ and any $y \notin V(G_i)$ such that $\mathbf{d}_{\mathbf{X_i}}(y) = \mathbf{t}$, in fact $x \in V(G_{i_1})$ and S resolves the pair $\langle x, y\rangle$ as it is a solution for I; and since $\mathbf{d}_{\mathbf{X_{i_1}}}(x) = \mathbf{r}^-$ and $\mathbf{d}_{\mathbf{X_{i_1}}}(y) = \mathbf{t}^-$, it follows by construction that $\langle \mathbf{r}^-, \mathbf{t}^- \rangle \in D_{pair}(I_1)$, and thus, $\langle \mathbf{r}, \mathbf{t} \rangle \in P_1$. Second, assume that $\mathbf{r} = \mathbf{d_{X_i}}(v)$ (note that v is then the only vertex in G_i with distance vector $\mathbf{r}$ to X_i). Let $y \notin V(G_i)$ be such that $\mathbf{d_{X_i}}(y) = \mathbf{t}$. Then, since S is a solution for I, there exists $s \in S$ such that s resolves the pair $\langle x, y \rangle$, which implies that $\mathbf{d_{X_{i_1}}}(s)$ resolves $\langle \mathbf{d_{X_i}}(v)^-, \mathbf{t}^- \rangle$, and thus, $\langle \mathbf{r}, \mathbf{t} \rangle \in C_1$.

(I5) Consider $\mathbf{r} \in [\mathtt{diam}(G)]^{|X_i|}$ for which there exists $x \in V(G_{i_1})$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, and assume that no vector in $D_{ext}(I)$ resolves the pair $\langle x, v \rangle$. Then, S must resolve the pair $\langle x, v \rangle$ as it is a solution for I; and since this holds for any vertex with distance vector $\mathbf{r}$ with respect to X_i, it follows by construction that $\langle \mathbf{r}^-, \mathbf{d_{X_{i_1}}}(v) \rangle \in D_{pair}(I_1)$. ◁

▷ Claim 49. S is a solution for I_1.

Proof. Let us prove that the conditions of Definition 33 are satisfied.

(S1) Let $\langle x, y \rangle$ be a pair of vertices of G_{i_1} and suppose that S does not resolve the pair $\langle x, y \rangle$ (we are done otherwise). Then, since S is a solution for I, there exists $\mathbf{r} \in D_{ext}(I)$ such that $\mathbf{r}$ resolves $\langle x, y \rangle$; but then, $\mathbf{r}^- \in D_{ext}(I_1)$ resolves $\langle x, y \rangle$.

(S2) Readily follows from the fact that $D_{int}(I_1) = \mathbf{d_{X_{i_1}}}(S)$.

(S3) By construction, for any $\langle \mathbf{r}, \mathbf{t} \rangle \in D_{pair}(I_1)$, any $x \in V(G_{i_1})$ such that $\mathbf{d_{X_{i_1}}}(x) = \mathbf{r}$, and any $y \notin V(G_{i_1})$ such that $\mathbf{d_{X_{i_1}}}(y) = \mathbf{t}$, S resolves the pair $\langle x, y \rangle$.

(S4) By construction, $S_{I_1} = S_I$, and thus, $S \cap X_{i_1} = S \cap (X_i \setminus \{v\}) = S_I = S_{I_1}$ as $v \notin S_I$ by assumption. ◁

The lemma now follows from the above two claims. ◀

▶ **Lemma 50.** *Let S be a minimum-size solution for I such that $v \in S$. Then, there exists $I_1 \in \mathcal{F}_2(I)$ such that $S \setminus \{v\}$ is a solution of I_1.*

Proof. Let I_1 be the instance for i_1 defined as follows.

- $S_{I_1} = S_I \setminus \{v\}$, $D_{ext}(I_1) = \{\mathbf{d_{X_{i_1}}}(v)\} \cup \{\mathbf{r}^- \mid \mathbf{r} \in D_{ext}(I)\}$, and $D_{int}(I_1) = \mathbf{d_{X_{i_1}}}(S \setminus \{v\})$.
- For every $\langle \mathbf{r}, \mathbf{t} \rangle \in ([\mathtt{diam}(G)]^{|X_{i_1}|})^2$, let $R_{\langle \mathbf{r}, \mathbf{t} \rangle} = \{\langle x, y \rangle \in V(G_{i_1}) \times V(G - G_{i_1}) \mid \mathbf{d_{X_{i_1}}}(x) = \mathbf{r}$ and $\mathbf{d_{X_{i_1}}}(y) = \mathbf{t}\}$. If $S \setminus \{v\}$ resolves every pair in $R_{\langle \mathbf{r}, \mathbf{t} \rangle}$, then we add $\langle \mathbf{r}, \mathbf{t} \rangle$ to $D_{pair}(I_1)$.

Let us prove that $I_1 \in \mathcal{F}_2(I)$ and that $S \setminus \{v\}$ is a solution of I_1. In the following, we let $S' = S \setminus \{v\}$.

▷ Claim 51. The constructed instance I_1 is compatible with I of type 2.

Proof. It is clear that conditions **(I'1)** and **(I'2)** of Definition 42 hold; let us show that the remaining conditions hold as well.

(I'3) Since S is a solution for I, for every $\mathbf{r} \in D_{int}(I) \setminus \{\mathbf{d_{X_i}}(v)\}$, there exists $s \in S$ such that $\mathbf{d_{X_i}}(s) = \mathbf{r}$ (in particular, $s \neq v$); but then, $\mathbf{r}^- = \mathbf{d_{X_{i_1}}}(s)$ and $\mathbf{r}^- \in D_{int}(I_1)$ by construction.

(I'4) Consider $\langle \mathbf{r}, \mathbf{t} \rangle \in D_{pair}(I)$. Let $x \in V(G_i)$ be such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$ and let $y \notin V(G_i)$ be such that $\mathbf{d_{X_i}}(y) = \mathbf{t}$. If v resolves the pair $\langle x, y \rangle$, then $\mathbf{r}_k = d(x, v) \neq d(y, v) = \mathbf{t}_k$, and so, $\langle \mathbf{r}, \mathbf{t} \rangle \in C_2$. Suppose therefore that v does not resolve $\langle x, y \rangle$. Then, since S is a solution for I, it must be that $S \setminus \{v\}$ resolves the pair $\langle x, y \rangle$; and since this holds for any pair with distance vectors $\langle \mathbf{r}, \mathbf{t} \rangle$ to X_i, $\langle \mathbf{r}^-, \mathbf{t}^- \rangle \in D_{pair}(I_1)$ by construction, and so, $\langle \mathbf{r}, \mathbf{t} \rangle \in P_2$. ◁

▷ Claim 52. S' is a solution for I_1.

Proof. Let us prove that the conditions of Definition 33 are satisfied.

(S1) Let $\langle x, y\rangle$ be a pair of vertices of G_{i_1} and suppose that S' does not resolve $\langle x, y\rangle$ (we are done otherwise). Then, since S is a solution for I, either $S \setminus S' = \{v\}$ resolves $\langle x, y\rangle$, in which case $\mathbf{d}_{\mathbf{X}_{\mathbf{i_1}}}(v) \in D_{ext}(I_1)$ resolves $\langle x, y\rangle$; or there exists a vector $\mathbf{r} \in D_{ext}(I)$ resolving $\langle x, y\rangle$, in which case $\mathbf{r}^- \in D_{ext}(I_1)$ resolves the pair as well.

(S2) Readily follows from the fact that $D_{int}(I_1) = \mathbf{d}_{\mathbf{X}_{\mathbf{i_1}}}(S')$.

(S3) By construction, for every $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I_1)$, any $x \in V(G_{i_1})$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i_1}}}(x) = \mathbf{r}$, and any $y \notin V(G_{i_1})$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i_1}}}(y) = \mathbf{t}$, S' resolves the pair $\langle x, y\rangle$.

(S4) By construction, $S_{I_1} = S_I \setminus \{v\}$, and so, $S \cap X_{i_1} = S \cap (X_i \setminus \{v\}) = S_I \setminus \{v\}$. ◁

The lemma now follows from the above two claims. ◀

As a consequence of Lemmas 47 and 50, the following holds.

▶ **Lemma 53.** *Let I be an instance for an introduce node i. Then,*

$$\dim(I) \geq \min\{\min_{I_1 \in \mathcal{F}_1(I)} \{\dim(I_1)\}, \min_{I_2 \in \mathcal{F}_2(I)} \{\dim(I_2) + 1\}\}.$$

Forget node. Let I be an instance for a forget node i with child i_1, and let $v \in V(G)$ be such that $X_i = X_{i_1} \setminus \{v\}$. Further, let $X_{i_1} = \{v_1, \ldots, v_k\}$, where $v = v_k$.

▶ **Definition 54.** *An instance I_1 for i_1 is compatible with I if the following hold.*

- **(F1)** $S_I = S_{I_1} \setminus \{v\}$.
- **(F2)** *For all* $\mathbf{r} \in D_{ext}(I_1)$*, there exists* $\mathbf{t} \in D_{ext}(I)$ *such that* $\mathbf{r}^- = \mathbf{t}$*.*
- **(F3)** *For all* $\mathbf{r} \in D_{int}(I)$*, there exists* $\mathbf{t} \in D_{int}(I_1)$ *such that* $\mathbf{t}^- = \mathbf{r}$*.*
- **(F4)** *For all* $\mathbf{r}, \mathbf{t} \in [\mathtt{diam}(G)]^{|X_i|}$*, let* $R_{\langle \mathbf{r},\mathbf{t}\rangle} = \{\langle x, y\rangle \in V(G_i) \times V(G - G_i) \mid \mathbf{d}_{\mathbf{X_i}}(x) = \mathbf{r}$ *and* $\mathbf{d}_{\mathbf{X_i}}(y) = \mathbf{t}\}$*. Then,* $D_{pair}(I) \subseteq \{\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_i|})^2 \mid \forall (x, y) \in R_{\langle \mathbf{r},\mathbf{t}\rangle}, \langle \mathbf{r}|d(x, v), \mathbf{t}|d(y, v)\rangle \in D_{pair}(I_1)\}$*.*

We denote by $\mathcal{F}_F(I)$ the set of instances for i_1 compatible with I. We aim to prove the following.

▶ **Lemma 55.** *Let I be an instance for a forget node i. Then,*

$$\dim(I) = \min_{I_1 \in \mathcal{F}_F(I)} \{\dim(I_1)\}.$$

To prove Lemma 55, we prove the following lemmas.

▶ **Lemma 56.** *Let I_1 be an instance for i_1 compatible with I, and let S be a minimum-size solution for I_1. Then, S is a solution for I. In particular,*

$$\dim(I) \leq \min_{I_1 \in \mathcal{F}_F(I)} \{\dim(I_1)\}.$$

Proof. Let us prove that the conditions of Definition 33 are satisfied.
(S1) Let $\langle x, y\rangle$ be a pair of vertices of G_i. Since $V(G_i) = V(G_{i_1})$ and S is a solution for I_1, either S resolves the pair $\langle x, y\rangle$, in which case we are done; or there exists $\mathbf{t} \in D_{ext}(I_1)$

such that $\mathbf{t}$ resolves $\langle x, y\rangle$. In the latter case, since I_1 is compatible with I, there then exists $\mathbf{r} \in D_{ext}(I)$ such that $\mathbf{t}^- = \mathbf{r}$; but then, $\mathbf{r}$ resolves the pair $\langle x, y\rangle$.

(S2) Consider $\mathbf{r} \in D_{int}(I)$. Since I_1 is compatible with I, there exists $\mathbf{t} \in D_{int}(I_1)$ such that $\mathbf{t}^- = \mathbf{r}$; and since S is a solution for I, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s) = \mathbf{t}$. Now, note that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(s) = \mathbf{t}^- = \mathbf{r}$.

(S3) Consider $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$. Let $x \in V(G_i)$ be such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(x) = \mathbf{r}$ and let $y \notin V(G_i)$ be such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(y) = \mathbf{t}$. Then, since I_1 is compatible with I, there exists $\langle \mathbf{u}, \mathbf{v}\rangle \in D_{pair}(I_1)$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{u}$ and $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(y) = \mathbf{v}$; and since S is a solution for I_1, S resolves the pair $\langle x, y\rangle$.

(S4) By compatibility, $S_I = S_{I_1} \setminus \{v\}$, and so, $S \cap X_i = S \cap (X_{i_1} \setminus \{v\}) = S_{I_1} \setminus \{v\}$. ◀

▶ **Lemma 57.** *Let S be a minimum-size solution for I. Then, there exists $I_1 \in \mathcal{F}_F(I)$ such that S is a solution for I_1. In particular,*

$$\dim(I) \geq \min_{I_1 \in \mathcal{F}_F(I)} \{\dim(I_1)\}.$$

Proof. Let I_1 be the instance for i_1 defined as follows.

- $S_{I_1} = S \cap X_{i_1}$, $D_{ext}(I_1) = \{\mathbf{r} \in [\mathtt{diam}(G)^{|X_{i_1}|}] \mid \mathbf{r}^- \in D_{ext}(I)$ and $\mathbf{r}_k = \min_{1\leq \ell\leq k-1}(\mathbf{r}^- + \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v))_\ell\}$, and $D_{int}(I_1) = \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(S)$.
- For every $\langle \mathbf{r}, \mathbf{t}\rangle \in ([\mathtt{diam}(G)]^{|X_{i_1}|})^2$, let $R_{\langle \mathbf{r},\mathbf{t}\rangle} = \{\langle x, y\rangle \in V(G_{i_1}) \times V(G - G_{i_1}) \mid \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(y) = \mathbf{t}\}$. If S resolves every pair in $R_{\langle \mathbf{r},\mathbf{t}\rangle}$, then we add $\langle \mathbf{r}, \mathbf{t}\rangle$ to $D_{pair}(I_1)$.

Let us prove that $I_1 \in \mathcal{F}_F(I)$ and that S is a solution for I_1.

▷ Claim 58. The constructed instance I_1 is compatible with I.

Proof. It is clear that conditions **(F1)** and **(F2)** of Definition 54 hold; let us show that the remaining conditions hold as well.

(F3) Since S is a solution for I, for every $\mathbf{r} \in D_{int}(I)$, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(s) = \mathbf{r}$; but then, $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s)^- = \mathbf{d}_{\mathbf{X}_\mathbf{i}}(s)$, where $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s) \in D_{int}(I_1)$ by construction.

(F4) Consider $\langle \mathbf{r}, \mathbf{t}\rangle \in D_{pair}(I)$. Let $x \in V(G_i)$ be such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(x) = \mathbf{r}$ and let $y \notin V(G_i)$ be such that $\mathbf{d}_{\mathbf{X}_\mathbf{i}}(y) = \mathbf{t}$. Then, S resolves the pair $\langle x, y\rangle$ as it is a solution for I; and since this holds for every pair $\langle a, b\rangle \in V(G_i) \times V(G - G_i)$ such that $(\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(a), \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(b)) = (\mathbf{r}|d(x, v), \mathbf{t}|d(y, v))$, by construction $\langle \mathbf{r}|d(x, v), \mathbf{t}|d(y, v)\rangle \in D_{pair}(I_1)$. ◁

▷ Claim 59. S is a solution for I_1.

Proof. Let us prove that the conditions of Definition 33 hold.

(S1) Let $\langle x, y\rangle$ be a pair of vertices of G_{i_1}. Since $V(G_i) = V(G_{i_1})$ and S is a solution for I, either S resolves the pair $\langle x, y\rangle$, in which case we are done: or there exists $\mathbf{t} \in D_{ext}(I)$ such that $\mathbf{t}$ resolves $\langle x, y\rangle$. In the latter case, by construction $\mathbf{t}|a \in D_{ext}(I_1)$, where $a = \min_{1\leq \ell\leq k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v))_\ell$; but then, $\mathbf{t}|a$ resolves $\langle x, y\rangle$.

(S2) Readily follows from the fact that $D_{int}(I_1) = \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(S)$.

(S3) By construction, for every $\langle \mathbf{r}, \mathbf{t} \rangle \in D_{pair}(I_1)$, any $x \in V(G_{i_1})$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}$, and any $y \notin V(G_{i_1})$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(y) = \mathbf{t}$, S resolves the pair $\langle x, y \rangle$.

(S4) By construction, $S_{I_1} = S \cap X_{i_1}$. ◁

The lemma now follows from the above two claims. ◀

To complete the proof of Theorem 26, let us now explain how the algorithm proceeds. Given a nice tree decomposition $(T, \mathcal{X})$ of a graph G rooted at node $r \in V(T)$, the algorithm computes the extended metric dimension for all possible instances in a bottom-up traversal of T. It computes the values for leaf nodes using Lemma 35, for join nodes using Lemma 37, for introduce nodes using Lemma 43, and for forget nodes using Lemma 55. The correctness of this algorithm follows from these lemmas and the following.

▶ **Lemma 60.** *Let G be a graph and let $(T, \{X_i : i \in V(T)\})$ be a nice tree decomposition of G rooted at node $r \in V(T)$. Then,*

$$\mathrm{md}(G) = \min_{S_r \subseteq X_r} \dim(X_r, S_r, \emptyset, \emptyset, \emptyset).$$

Proof. Let S be a minimum-size resolving set of G. Then, by Definition 33, S is a solution for the EMD instance $(X_r, S \cap X_r, \emptyset, \emptyset, \emptyset)$, and so,

$$\min_{S_r \subseteq X_r} \dim(X_r, S_r, \emptyset, \emptyset, \emptyset) \leq \dim(X_r, S \cap X_r, \emptyset, \emptyset, \emptyset) \leq \mathrm{md}(G).$$

Conversely, let $S' \subseteq X_r$ be a set attaining the minimum above, and let S be a minimum-size solution for the EMD instance $(X_r, S', \emptyset, \emptyset, \emptyset)$. Then, by Definition 33, every vertex of $G_r = G$ is resolved by S, and so,

$$\mathrm{md}(G) \leq \dim(X_r, S', \emptyset, \emptyset, \emptyset) = \min_{S_r \subseteq X_r} \dim(X_r, S_r, \emptyset, \emptyset, \emptyset),$$

which concludes the proof. ◀

To get the announced complexity, observe first that, at each node $i \in V(T)$, there are at most $2^{|X_i|} \cdot 2^{\mathtt{diam}(G)^{|X_i|}} \cdot 2^{\mathtt{diam}(G)^{|X_i|}} \cdot 2^{\mathtt{diam}(G)^{2|X_i|}}$ possible instances to consider, where $|X_i| = \mathcal{O}(\mathtt{tw}(G))$. Since T has $\mathcal{O}(\mathtt{tw}(G) \cdot n)$ nodes, there are in total $\mathcal{O}(\alpha(\mathtt{tw}(G)) \cdot \mathtt{tw}(G) \cdot n)$ possible instances, where $\alpha(k) = 2^k \cdot 2^{\mathtt{diam}(G)^k} \cdot 2^{\mathtt{diam}(G)^k} \cdot 2^{\mathtt{diam}(G)^{2k}}$. The running time of the algorithm then follows from these facts and the next lemma (note that to avoid repeated computations, we can first compute the distance between every pair of vertices of G in $n^{\mathcal{O}(1)}$ time, as well as all possible distance vectors to a bag from the possible distance vectors to its child/children).

▶ **Lemma 61.** *Let I be an* EMD *instance for a node $i \in V(T)$, and assume that, for every child i_1 of i and every* EMD *instance I_1 for i_1 compatible with I, $\dim(I_1)$ is known. Then, $\dim(I)$ can be computed in time $\mathcal{O}(\alpha(|X_i|)) \cdot n^{\mathcal{O}(1)}$.*

Proof. If i is a leaf node, then $\dim(I)$ can be computed in constant time by Lemma 35. Otherwise, let us prove that one can compute all compatible instances in the child nodes in the announced time (note that i has at most two child nodes). First, given a 5-tuple $(X_{i_1}, S_{I_1}, D_{int}(I_1), D_{ext}(I_1), D_{pair}(I_1))$, checking whether it is an EMD instance can be done in $\mathcal{O}(|I_1|) \cdot n^{\mathcal{O}(1)}$ time; and the number of such 5-tuples is bounded by $\alpha(|X_{i_1}|)$. It is also not difficult to see that checking for compatibility can, in each case, be done in $\mathcal{O}(|I|) \cdot n^{\mathcal{O}(1)}$ time. Now, note that, by Definition 33, $|I| = \mathcal{O}(\mathtt{diam}(G)^{\mathcal{O}(|X_i|)})$ and thus, computing all compatible instances can indeed be done in $\mathcal{O}(\alpha(|X_i|)) \cdot n^{\mathcal{O}(1)}$. Then, since computing the minimum using the formulas of Lemmas 37, 43, and 55 can be done in $\mathcal{O}(\alpha(|X_i|)$ time, the lemma follows. ◀

8.2 Dynamic Programming Algorithm for Geodetic Set

In this subsection, we prove the following theorem.

▶ **Theorem 62.** *Geodetic Set admits an algorithm running in time* $2^{\mathtt{diam}^{\mathcal{O}(\mathtt{tw})}} \cdot n^{\mathcal{O}(1)}$.

The proof follows along the same lines as that of the proof of Theorem 26.

Overview. We first give an intuitive description of the dynamic programming scheme. At each step of the algorithm, we consider a bounded number of *solution types*, depending on the properties of the solution vertices with respect to the current bag. At a given dynamic programming step, we will assume that the current solution covers all vertices in G_i. Such a vertex may be covered by (1) two vertices in G_i, (2) a vertex in G_i and a vertex in $G - G_i$, or (3) two vertices in $G - G_i$.

Any bag X_i of the tree decomposition whose node i lies on a path between two join nodes in T, forms a separator of G: there are no edges between the vertices of $G_i - X_i$ and $G - G_i$. For a vertex v not in X_i, we consider its distance-vector to the vertices of X_i; the distance-vectors induce an equivalence relation on the vertices of $G - X_i$, whose classes we call X_i*-classes*. Consider the two subgraphs G_i and $G - G_i$. Given a vertex z in G_i, any two solution vertices x, y from $G - G_i$ that are in the same X_i-class, will cover together with z the exact same vertices from G_i, that is, a vertex u of G_i is covered by z and x if and only if it is covered by z and y. Thus, for case (2), it is irrelevant whether x or y will be in a geodetic set, and it is sufficient to know that a vertex of their X_i-class will eventually be chosen. Similarly, for any four vertices x_1, x_2, y_1, y_2 of $G - G_i$ such that x_1, x_2 (y_1, y_2, respectively) are in the same X_i-class *and* $d(x_1, y_1) = d(x_2, y_2)$, we have that x_1 and y_1 cover exactly the same vertices in G_i as x_2 and y_2. Thus, for case (3), it is irrelevant whether x_1, y_1 or x_2, y_2 will be in the geodetic set, and it is sufficient to know that a vertex from each of their X_i-classes whose distance between them is $d(x_1, y_1)$ will eventually be chosen.

The same idea is used to "remember" the previously computed solution: it is sufficient to remember the X_i-classes of the vertices in the previously computed geodetic set, as well as pairs C_1, C_2 of X_i-classes together with an integer d corresponding to the distance between any vertex in C_1 and any vertex in C_2, rather than the vertices themselves.

Keeping track, in the aforementioned way, of the "past" and "future" solution, is sufficient when processing a join node i. Indeed, for a join node i with children i_1,i_2, a vertex of $G_i = G_{i_1} \cup G_{i_2}$ may be covered by: two vertices from $G - G_i$; a vertex from G_{i_1} and a vertex from $G - G_i$; two vertices from G_{i_1}; a vertex from G_{i_1} and a vertex from G_{i_2}; two vertices from G_{i_2}; a vertex from G_{i_2} and a vertex from $G - G_i$. This is also sufficient when processing an introduce node i where a new vertex v is introduced (i.e., added to the child bag $X_{i'}$ to form X_i). Indeed, we may check that v is either covered by two vertices from G_{i_1}; a vertex of G_{i_1} and a vertex of $G - G_i$; or two vertices of $G - G_i$. If this does not hold, then we add v into the solution.

For a bag X_i and a vertex v not in X_i, the number of possible distance vectors to the vertices of X_i is at most $\mathtt{diam}(G)^{|X_i|}$. Thus, a solution for bag X_i will consist of: (i) the subset of vertices of X_i selected in the solution; (ii) a subset of the $\mathtt{diam}(G)^{|X_i|}$ possible vectors to denote the X_i-classes from which the currently computed solution (for G_i) contains at least one vertex in the geodetic set; (iii) a subset of the $\mathtt{diam}(G)^{|X_i|}$ possible vectors denoting the X_i-classes from which the future solution needs at least one vertex of $G - G_i$ in the geodetic set; (iv) a subset of the $\mathtt{diam}(G)^{|X_i|} \times \mathtt{diam}(G)^{|X_i|} \times \mathtt{diam}(G)$ possible elements representing the pairs of X_i-classes and their distance to each other from which the currently computed solution (for G_i) contains at least two vertices in the geodetic set; (v) a subset of the $\mathtt{diam}(G)^{|X_i|} \times \mathtt{diam}(G)^{|X_i|} \times \mathtt{diam}(G)$ possible elements representing the pairs of X_i-classes and their distance from which the future solution needs at least two vertices of $G - G_i$ in the geodetic set.

Formal description. Before presenting the dynamic program, we first introduce some useful definitions and lemmas (see also Section 8.1 for missing definitions).

For a set S, we denote by $\mathcal{P}_2(S)$ the set of subsets of S of size 2. Given a graph G and a set $S \subseteq V(G)$, we say that a vertex $x \in V(G)$ is *covered by* S if either $x \in S$ or there exist $u, v \in S$ such that x lies on a shortest path from u to v. The smallest size of a geodetic set for G is denoted by $\mathrm{gs}(G)$.

▶ **Definition 63.** *Let $\mathbf{r_1}, \mathbf{r_2}$, and $\mathbf{r_3}$ be three vectors of size k, and let d be an integer. We say that $\mathbf{r_3}$ is* covered *by $(\{\mathbf{r_1}, \mathbf{r_2}\}, d)$ if*

$$\min_{1 \le i \le k} (\mathbf{r_1} + \mathbf{r_3})_i + \min_{1 \le i \le k} (\mathbf{r_2} + \mathbf{r_3})_i = d.$$

▶ **Definition 64.** *Let G be a graph and let $X = \{v_1, \dots, v_k\}$ be a subset of vertices of G. Given a vertex x of G, the* distance vector *$\mathbf{d_X}(x)$ of x to X is the vector of size k such that for all $1 \le j \le k$, $\mathbf{d_X}(x)_j = d(x, v_j)$. For a set $S \subseteq V(G)$, we let $\mathbf{d_X}(S) = \{\mathbf{d_X}(s) \mid s \in S\}$.*

▶ **Definition 65.** *Let G be a graph and let $X = \{v_1, \dots, v_k\}$ be a subset of vertices of G.*

- *Let $\mathbf{r_1}, \mathbf{r_2}$ be two vectors of size k and let d be an integer. Then, for any $x \in V(G)$, we say that x is* covered *by $(\{\mathbf{r_1}, \mathbf{r_2}\}, d)$ if $\mathbf{d_X}(x)$ is covered by $(\{\mathbf{r_1}, \mathbf{r_2}\}, d)$.*
- *Let x, y be two vertices of G and let $\mathbf{r}$ be a vector of size k. We say that $\mathbf{r}$ is* covered *by x and y if $\mathbf{r}$ is covered by $(\{\mathbf{d_X}(x), \mathbf{d_X}(y)\}, d(x, y))$. More generally, given a set S of vertices of G, we say that $\mathbf{r}$ is* covered *by S if there exist $x, y \in S$ such that $\mathbf{r}$ is covered by x and y.*
- *Let s be a vertex of G and let $\mathbf{r}$ be a vector of size k. Then, for any $x \in V(G)$, we say that x is* covered *by s and $\mathbf{r}$ if $d(s, x) + \min_{1 \le j \le k} (\mathbf{d_X}(x) + \mathbf{r})_j = \min_{1 \le j \le k} (\mathbf{d_X}(s) + \mathbf{r})_j$.*

▶ **Lemma 66.** *Let $X = \{v_1, \dots, v_k\}$ be a separator of a graph G, and let G_1 be a connected component of $G - X$. Further, let $x \in V(G_1) \cup X$.*

(1) *Let $\mathbf{r_1}, \mathbf{r_2}$ be two vectors of size k, and let d be an integer. If x is covered by $(\{\mathbf{r_1}, \mathbf{r_2}\}, d)$, then, for any $u, v \in V(G - G_1)$ such that $\mathbf{d_X}(u) = \mathbf{r_1}$, $\mathbf{d_X}(v) = \mathbf{r_2}$, and $d(u, v) = d$, x is covered by u and v.*

(2) *Let s be a vertex of $V(G_1) \cup X$ and let $\mathbf{r}$ be a vector of size k. If x is covered by s and $\mathbf{r}$, then, for any $u \in V(G - G_1)$ such that $\mathbf{d_X}(u) = \mathbf{r}$, x is covered by s and u.*

(3) *Let $\mathbf{r_1}, \mathbf{r_2}$ be two vectors of size k. If x is covered by $(\{\mathbf{r_1}, \mathbf{r_2}\}, \min_{1 \le j \le k} (\mathbf{r_1} + \mathbf{r_2})_j)$, then, for any $u \in V(G - G_1)$ such that $\mathbf{d_X}(u) = \mathbf{r_1}$, x is covered by u and $\mathbf{r_2}$.*

(4) *Let $\mathbf{r}$ be a vector of size k and let u, v be two vertices of $G - G_1$. If $\mathbf{r}$ is covered by u and v, then, for any $w \in V(G_1) \cup X$ such that $\mathbf{d_X}(w) = \mathbf{r}$, w is covered by u and v.*

Proof. To prove item (1), it suffices to note that since X separates x from u, $d(x, u) = \min_{1 \le j \le k} (\mathbf{d_X}(x) + \mathbf{d_X}(u))_j$ (note that if x or u belongs to X, then surely this equality holds as well); and for the same reason, $d(x, v) = \min_{1 \le j \le k} (\mathbf{d_X}(x) + \mathbf{d_X}(v))_j$. Now, x is covered by $(\{\mathbf{d_X}(u), \mathbf{d_X}(v)\}, d(u, v))$, and so, $d(u, x) + d(x, v) = d(u, v)$ by definition. Items (2), (3), and (4) follow from similar arguments. ◀

We now define the problem solved at each step of the dynamic programming algorithm, called Extended Geodetic Set (EGS for short), whose instances are defined as follows.

▶ **Definition 67.** *Let G be a graph and let $(T, \{X_i : i \in V(T)\})$ be a tree decomposition of G. For a node i of T, an* instance *of* EGS *is a 6-tuple $I = (X_i, S_I, D_{int}(I), D_{ext}(I), D_{int/int}(I), D_{ext/ext}(I))$ composed of the bag X_i of i, a subset S_I of X_i, and four sets satisfying the following.*

- $D_{int}(I), D_{ext}(I) \subseteq [\mathtt{diam}(G)]^{|X_i|}$.
- $D_{int/int}(I), D_{ext/ext}(I) \subseteq \mathcal{P}_2([\mathtt{diam}(G)]^{|X_i|}) \times [\mathtt{diam}(G)]$.
- *For each $\mathbf{r} \in D_{ext}(I)$, there exists $x \notin V(G_i)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$.*
- *For each $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{ext/ext}(I)$, there exist $x, y \notin V(G_i)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, $\mathbf{d_{X_i}}(y) = \mathbf{t}$ and $d(x, y) = d$.*
- $S_I \neq \emptyset$ *or* $D_{ext}(I) \neq \emptyset$ *or* $D_{ext/ext}(I) \neq \emptyset$.

▶ **Definition 68.** *A set $S \subseteq V(G_i)$ is a solution for an instance I of* EGS *if the following hold.*

- ***(S1)*** *Every vertex of G_i is either covered by S, covered by a vertex in S and a vector in $D_{ext}(I)$, or covered by an element of $D_{ext/ext}(I)$.*
- ***(S2)*** *For each $\mathbf{r} \in D_{int}(I)$, there exists $s \in S$ such that $\mathbf{d_{X_i}}(s) = \mathbf{r}$.*
- ***(S3)*** *For each $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I)$, there exist two distinct vertices $s_1, s_2 \in S$ such that $\mathbf{d_{X_i}}(s_1) = \mathbf{r_1}$, $\mathbf{d_{X_i}}(s_2) = \mathbf{r_2}$, and $d(s_1, s_2) = d$.*
- ***(S4)*** $S \cap X_i = S_I$.

In the remainder of this section, for brevity, we will refer to an instance of the EGS problem only as an instance.

▶ **Definition 69.** *Let I be an instance. We denote by $\dim(I)$ the minimum size of a set $S \subseteq V(G_i)$ which is a solution for I. If no such set exists, then we set $\dim(I) = +\infty$. We refer to this value as the* extended geodetic set number *of I.*

In the following, we fix a graph G and a nice tree decomposition $(T, \{X_i : i \in V(T)\})$ of G. Given a node i of T and an instance I for i, we show how to compute $\dim(I)$. The proof is divided according to the type of the node i.

Leaf node. Computing $\dim(I)$ when I is an instance for a leaf node can be done with the following lemma.

▶ **Lemma 70.** *Let I be an instance for a leaf node i and let v be the only vertex in X_i. Then,*

$$\dim(I) = \begin{cases} 0 & \text{if } S_I = \emptyset, D_{int}(I) = \emptyset, \text{ and } D_{int/int}(I) = \emptyset \\ 1 & \text{if } S_I = \{v\}, D_{int}(I) \subseteq \{(0)\}, \text{ and } D_{int/int}(I) = \emptyset \\ +\infty & \text{otherwise} \end{cases}$$

Proof. Suppose first that $S_I = \emptyset$. Then, the empty set is the only possible solution for I; and the empty set is a solution for I only if $D_{int}(I) = \emptyset$ and $D_{int/int}(I) = \emptyset$. Suppose next that $S_I = \{v\}$. Then, the set $S = \{v\}$ is the only possible solution for I; and this set is a solution for I only if $D_{int}(I) = \emptyset$ or $D_{int}(I)$ contains only the vector $\mathbf{d_{X_i}}(v) = (0)$, and $D_{int/int}(I) = \emptyset$. ◀

In the remainder of this section, we handle the three other types of nodes. For each type of node, we proceed as follows: we first define a notion of compatibility on the instances for the child/children of a node i and show how to compute the extended geodetic set number of an instance I for i from the extended geodetic set number of instances for the child/children of i compatible with I.

Join node. Let I be an instance for a join node i, and let i_1 and i_2 be the two children of i. In the following, we let $X_i = \{v_1, \ldots, v_k\} = X_{i_1} = X_{i_2}$.

▶ **Definition 71.** *A pair of instances (I_1, I_2) for (i_1, i_2) is* compatible *with I if the following hold.*

- ***(J1)*** $S_{I_1} = S_{I_2} = S_I$.
- ***(J2)*** $D_{int}(I) \subseteq D_{int}(I_1) \cup D_{int}(I_2)$.
- ***(J3)*** $D_{ext}(I_1) \subseteq D_{ext}(I) \cup D_{int}(I_2)$ *and* $D_{ext}(I_2) \subseteq D_{ext}(I) \cup D_{int}(I_1)$.
- ***(J4)*** *Let* $D_1 = \{(\{\mathbf{r_1}, \mathbf{r_2}\}, \min_{1\le j\le k}(\mathbf{r_1} + \mathbf{r_2})_j) \mid \mathbf{r_1} \in D_{ext}(I), \mathbf{r_2} \in D_{int}(I_2)\}$. *Then,* $D_{ext/ext}(I_1) \subseteq D_{ext/ext}(I) \cup D_{int/int}(I_2) \cup D_1$.
  Symmetrically, let $D_2 = \{(\{\mathbf{r_1}, \mathbf{r_2}\}, \min_{1\le j\le k}(\mathbf{r_1} + \mathbf{r_2})_j) \mid \mathbf{r_1} \in D_{ext}(I), \mathbf{r_2} \in D_{int}(I_1)\}$. *Then,* $D_{ext/ext}(I_2) \subseteq D_{ext/ext}(I) \cup D_{int/int}(I_1) \cup D_2$.
- ***(J5)*** *Let* $F = \{(\{\mathbf{r_1}, \mathbf{r_2}\}, \min_{1\le j\le k}(\mathbf{r_1} + \mathbf{r_2})) \mid \mathbf{r_1} \in D_{int}(I_1), \mathbf{r_2} \in D_{int}(I_2)\}$. *Then,* $D_{int/int}(I) \subseteq D_{int/int}(I_1) \cup D_{int/int}(I_2) \cup F$.

Let $\mathcal{F}_J(I)$ be the set of pairs of instances compatible with I. We aim to prove the following.

▶ **Lemma 72.** *Let I be an instance for a join node i. Then,*

$$\dim(I) = \min_{(I_1,I_2)\in\mathcal{F}_J(I)} (\dim(I_1) + \dim(I_2) - |S_I|).$$

To prove Lemma 72, we prove the following two lemmas.

▶ **Lemma 73.** *Let (I_1, I_2) be a pair of instances for (i_1, i_2) compatible with I such that $\dim(I_1)$ and $\dim(I_2)$ have finite values. Let S_1 be a minimum-size solution for I_1 and S_2 a minimum-size solution for I_2. Then, $S = S_1 \cup S_2$ is a solution for I. In particular,*

$$\dim(I) \le \min_{(I_1,I_2)\in\mathcal{F}_J(I)} (\dim(I_1) + \dim(I_2) - |S_I|).$$

Proof. Let us show that every condition of Definition 68 is satisfied.

(S1) Let x be a vertex of G_i and assume, without loss of generality, that $x \in V(G_{i_1})$ (the case where $x \in V(G_{i_2})$ is symmetric). Then, since S_1 is a solution for I_1, either x is covered by S_1, in which case we are done; or (1) x is covered by a vertex $s \in S_1$ and a vector $\mathbf{r} \in D_{ext}(I_1)$; or (2) x is covered by an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$.

Suppose first that (1) holds. Then, by compatibility, either $\mathbf{r} \in D_{ext}(I)$, in which case we are done, or $\mathbf{r} \in D_{int}(I_2)$. In the latter case, since S_2 is a solution for I_2, there exists $s_2 \in S_2$ such that $\mathbf{r} = \mathbf{d_{X_{i_2}}}(s_2) = \mathbf{d_{X_i}}(s_2)$; but then, by Lemma 66(2), x is covered by $s_1, s_2 \in S$.

Suppose next that (2) holds. Then, by compatibility, either $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I)$, in which case we are done; or (i) $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I_2)$; or (ii) $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_1$. Now, if (i) holds, then, since S_2 is a solution for I_2, there exist $s_1, s_2 \in S_2$ such that $\mathbf{r_1} = \mathbf{d_{X_{i_2}}}(s_1) = \mathbf{d_{X_i}}(s_1)$, $\mathbf{r_2} = \mathbf{d_{X_{i_2}}}(s_2) = \mathbf{d_{X_i}}(s_2)$, and $d = d(s_1, s_2)$; but then, by Lemma 66(1), x is covered by $s_1, s_2 \in S$. Thus, suppose that (ii) holds. Then, since S_2 is a solution for I_2, there exists $s_2 \in S_2$ such that, say, $\mathbf{r_2} = \mathbf{d_{X_{i_2}}}(s_2) = \mathbf{d_{X_i}}(s_2)$; but then, by Lemma 66(3), x is covered by $s_2 \in S$ and $\mathbf{r_1} \in D_{ext}(I)$.

(S2) Consider a vector $\mathbf{r} \in D_{int}(I)$. Then, by compatibility, $\mathbf{r} \in D_{int}(I_1) \cup D_{int}(I_2)$, say $\mathbf{r} \in D_{int}(I_1)$ without loss of generality. Now, S_1 is a solution for I_1, and so, there exists $s_1 \in S_1 \subseteq S$ such that $\mathbf{r} = \mathbf{d_{X_{i_1}}}(s_1) = \mathbf{d_{X_i}}(s_1)$.

(S3) Consider an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I)$. Then, by compatibility, $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I_1) \cup D_{int/int}(I_2) \cup F$. Now, if $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I_1)$, then, since S_1 is a solution for I_1, there exist $s_1, s_2 \in S_1 \subseteq S$ such that $\mathbf{r_1} = \mathbf{d_{X_{i_1}}}(s_1) = \mathbf{d_{X_i}}(s_1)$, $\mathbf{r_2} = \mathbf{d_{X_{i_1}}}(s_2) = \mathbf{d_{X_i}}(s_2)$, and $d(s_1, s_2) = d$; and we conclude symmetrically if $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I_2)$. Thus, suppose that $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in F$. Then, since S_1 and S_2 are solutions for I_1 and I_2, respectively, there exist $s_1 \in S_1$ and $s_2 \in S_2$ such that $\mathbf{r_1} = \mathbf{d_{X_{i_1}}}(s_1) = \mathbf{d_{X_i}}(s_1)$ and $\mathbf{r_2} = \mathbf{d_{X_{i_1}}}(s_2) = \mathbf{d_{X_i}}(s_2)$; but then, since X_i separates s_1 and s_2, $\min_{1 \le j \le k}(\mathbf{r_1} + \mathbf{r_2})_j = d(s_1, s_2)$.

(S4) By compatibility, $S_{I_1} = S_{I_2} = S_I$, and thus, $S \cap X_i = S \cap X_{i_1} = S_{I_1} = S_I$.

It now follows from the above that $\dim(I) \le |S| = |S_1| + |S_2| - |S_I| = \dim(I_1) + \dim(I_2) - |S_I|$; and since this holds true for any $(I_1, I_2) \in \mathcal{F}_J(I)$, the lemma follows. ◀

▶ **Lemma 74.** *Let I be an instance for a join node i. Then,*

$$\dim(I) \ge \min_{(I_1, I_2) \in \mathcal{F}_J(I)} (\dim(I_1) + \dim(I_2) - |S_I|).$$

Proof. If $\dim(I) = +\infty$, then the inequality readily holds. Thus, assume that $\dim(I) < +\infty$ and let S be a minimum-size solution for I. For $j \in \{1, 2\}$, let $S_j = S \cap V(G_{i_j})$. Now, let I_1 and I_2 be the two instances for i_1 and i_2, respectively, defined as follows.

- $S_{I_1} = S_{I_2} = S_I$.
- $D_{int}(I_1) = \mathbf{d_{X_i}}(S_1)$ and $D_{int}(I_2) = \mathbf{d_{X_i}}(S_2)$.
- $D_{ext}(I_1) = D_{ext}(I) \cup D_{int}(I_2)$ and $D_{ext}(I_2) = D_{ext}(I) \cup D_{int}(I_1)$.
- $D_{int/int}(I_1) = \{(\{\mathbf{d_{X_{i_1}}}(s_1), \mathbf{d_{X_{i_1}}}(s_2)\}, d(s_1, s_2)) \mid s_1, s_2 \in S_1\}$ and $D_{int/int}(I_2) = \{(\{\mathbf{d_{X_{i_1}}}(s_1), \mathbf{d_{X_{i_1}}}(s_2)\}, d(s_1, s_2)) \mid s_1, s_2 \in S_2\}$.
- $D_{ext/ext}(I_1) = D_{ext/ext}(I) \cup D_{int/int}(I_2) \cup D_1$ and $D_{ext/ext}(I_2) = D_{ext/ext}(I) \cup D_{int/int}(I_1) \cup D_2$ (see Definition 71 for the definitions of D_1 and D_2).

Let us show that the pair of instances (I_1, I_2) is compatible with I and that, for $j \in \{1, 2\}$, S_j is a solution for I_j.

▷ Claim 75. The constructed pair of instances (I_1, I_2) for (i_1, i_2) is compatible with I.

Proof. It is clear that conditions **(J1)** through **(J4)** of Definition 71 hold; let us show that condition **(J5)** holds as well.

(J5) Consider an element $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$. Since S is a solution for I, there exist $x, y \in S$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, $\mathbf{d_{X_i}}(y) = \mathbf{t}$, and $d(x, y) = d$. Now, if $x, y \in S_1$, then, by construction, $(\{\mathbf{d_{X_{i_1}}}(x), \mathbf{d_{X_{i_1}}}(y)\}, d(x, y)) \in D_{int/int}(I_1)$; and we conclude symmetrically if $x, y \in S_2$. Thus, assume, without loss of generality, that $x \in S_1$ and $y \in S_2$. Then, by construction, $\mathbf{d_{X_{i_1}}}(x) \in D_{int}(I_1)$ and $\mathbf{d_{X_{i_2}}}(y) \in D_{int}(I_2)$; and since X_i separates x and y, $d = d(x, y) = \min_{1 \le j \le k}(\mathbf{d_{X_i}}(x) + \mathbf{d_{X_i}}(y))_j$, that is, $(\{\mathbf{r}, \mathbf{t}\}, d) \in F$. ◁

▷ Claim 76. For every $j \in \{1, 2\}$, S_j is a solution for I_j.

Proof. We only prove that S_1 is a solution for I_1 as the other case is symmetric. To this end, let us show that every condition of Definition 68 is satisfied.

(S1) Consider a vertex x of G_{i_1}. Then, since $V(G_{i_1}) \subseteq V(G_i)$ and S is a solution for I, either (1) x is covered by two vertices $s_1, s_2 \in S$; or (2) x is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I)$; or (3) x is covered by an element of $D_{ext/ext}(I)$. Since, by construction, $D_{ext/ext}(I) \subseteq D_{ext/ext}(I_1)$, let us assume that (3) does not hold (we are done otherwise).

Suppose first that (1) holds and assume that at least one of s_1 and s_2 does not belong to S_1 (we are done otherwise), say $s_2 \notin S_1$ without loss of generality. Then, $\mathbf{d_{X_{i_2}}}(s_2) \in D_{int}(I_2) \subseteq D_{ext}(I_1)$ by construction, and thus, if $s_1 \in S_1$, then x is covered by $s_1 \in S_1$ and $\mathbf{d_{X_{i_1}}}(s_2) \in D_{ext}(I_1)$. Suppose therefore that $s_1, s_2 \in S_2$. Then, by construction, $(\{\mathbf{d_{X_i}}(s_1), \mathbf{d_{X_i}}(s_2)\}, d(s_1, s_2)) \in D_{int/int}(I_2)$; but $D_{int/int}(I_2) \subseteq D_{ext/ext}(I_1)$ by construction, and thus, x is covered by an element of $D_{ext/ext}(I_1)$.

Second, suppose that (2) holds. Then, by construction, $\mathbf{r} \in D_{ext}(I_1)$, and thus, if $s \in S_1$, then we are done. Now, if $s \in S_2$, then $\mathbf{d_{X_{i_2}}}(s) \in D_{int}(I_2)$ which implies that $(\{\mathbf{r}, \mathbf{d_{X_i}}(s)\}, \min_{1 \le j \le k}(\mathbf{r} + \mathbf{d_{X_i}}(s))_j) \in D_1$; but then, x is covered by an element of $D_{ext/ext}(I_1)$.

(S2) and **(S3)** readily follow from the fact that, by construction, $D_{int}(I_1) = \mathbf{d_{X_i}}(S_1)$ and $D_{int/int}(I_1) = \{(\{\mathbf{d_{X_{i_1}}}(s_1), \mathbf{d_{X_{i_1}}}(s_2)\}, d(s_1, s_2)) \mid s_1, s_2 \in S_1\}$, respectively.

(S4) By construction, $S_{I_1} = S_I$, and thus, $S \cap X_{i_1} = S \cap X_i = S_I = S_{I_1}$. ◁

To conclude, since the sets S_1 and S_2 are solutions for I_1 and I_2, respectively, $\dim(I_1) \le |S_1|$ and $\dim(I_2) \le |S_2|$. Now, $|S| = |S_1| + |S_2| - |S_I|$, and so, $\dim(I) = |S| \ge \dim(I_1) + \dim(I_2) - |S_I| \ge \min_{(J_1, J_2) \in \mathcal{F}_J(I)}(\dim(J_1) + \dim(J_2) - |S_I|)$. ◀

Introduce node. Let I be an instance for an introduce node i with child i_1, and let $v \in V(G)$ be such that $X_i = X_{i_1} \cup \{v\}$. In the following, we let $X_i = \{v_1, \dots, v_k\}$ where $v = v_k$.

▶ **Definition 77.** *An instance I_1 for i_1 is compatible with I of* type 1 *if the following hold.*

- **(I1)** $S_I = S_{I_1}$.
- **(I2)** *For each* $\mathbf{r} \in D_{int}(I)$, $\mathbf{r}_k = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_j$ *and* $\mathbf{r}^- \in D_{int}(I_1)$.
- **(I3)** *For each* $\mathbf{r} \in D_{ext}(I_1)$, *there exists* $\mathbf{t} \in D_{ext}(I)$ *such that* $\mathbf{t}^- = \mathbf{r}$.
- **(I4)** *For each* $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$, $\mathbf{r}_k = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_j$, $\mathbf{t}_k = \min_{1 \le j \le k-1}(\mathbf{t} + \mathbf{d_{X_{i_1}}}(v))_j$, *and* $(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \in D_{int/int}(I_1)$.
- **(I5)** *For each* $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$, *there exists* $(\{\mathbf{t_1}, \mathbf{t_2}\}, d) \in D_{ext/ext}(I)$ *such that* $\mathbf{t_1^-} = \mathbf{r_1}$ *and* $\mathbf{t_2^-} = \mathbf{r_2}$.
- **(I6)** *One of the following holds.*
  - v *is covered by an element of* $D_{int/int}(I_1)$.
  - *There exist* $\mathbf{r} \in D_{int}(I_1)$ *and* $\mathbf{t} \in D_{ext}(I)$ *such that* v *is covered by* $(\{\mathbf{r}|d, \mathbf{t}\}, \min_{1 \le j \le k}(\mathbf{r}|d + \mathbf{t})_j)$ *where* $d = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_j$.
  - v *is covered by an element of* $D_{ext/ext}(I)$.

An instance I_1 for i_1 is compatible with I of type 2 *if the following hold.*

- **(I'1)** $S_I = S_{I_1} \cup \{v\}$.
- **(I'2)** *For each* $\mathbf{r} \in D_{int}(I) \setminus \{\mathbf{d_{X_i}}(v)\}$, $\mathbf{r}_k = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_j$ *and* $\mathbf{r}^- \in D_{int}(I_1)$.
- **(I'3)** *For each* $\mathbf{r} \in D_{ext}(I_1) \setminus \{\mathbf{d_{X_{i_1}}}(v)\}$, *there exists* $\mathbf{t} \in D_{ext}(I)$ *such that* $\mathbf{t}^- = \mathbf{r}$.
- **(I'4)** *For each* $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$, $\mathbf{r}_k = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_j$ *and* $\mathbf{t}_k = \min_{1 \le j \le k-1}(\mathbf{t} + \mathbf{d_{X_{i_1}}}(v))_j$. *Furthermore, one of the following holds:*
  - $(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \in D_{int/int}(I_1)$,
  - $\mathbf{r} = \mathbf{d_{X_i}}(v)$, $d = \mathbf{t}_k$, *and* $\mathbf{t}^- \in D_{int}(I_1)$, *or*
  - $\mathbf{t} = \mathbf{d_{X_i}}(v)$, $d = \mathbf{r}_k$, *and* $\mathbf{r}^- \in D_{int}(I_1)$.
- **(I'5)** *For each* $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$, *one of the following holds:*
  - *there exist* $x, y \notin V(G_{i_1}) \cup \{v\}$ *such that* $\mathbf{d_{X_{i_1}}}(x) = \mathbf{r_1}$, $\mathbf{d_{X_{i_1}}}(y) = \mathbf{r_2}$, $d(x, y) = d$, *and* $(\{\mathbf{r_1}|d(x, v), \mathbf{r_2}|d(y, v)\}, d) \in D_{ext/ext}(I)$

- $\mathbf{r_1} = \mathbf{d_{X_{i_1}}}(v)$ *and there exists* $x \notin V(G_{i_1}) \cup \{v\}$ *such that* $\mathbf{d_{X_i}}(x) = \mathbf{r_2}|d$ *and* $\mathbf{r_2}|d \in D_{ext}(I)$*, or*
- $\mathbf{r_2} = \mathbf{d_{X_{i_1}}}(v)$ *and there exists* $x \notin V(G_{i_1}) \cup \{v\}$ *such that* $\mathbf{d_{X_i}}(x) = \mathbf{r_1}|d$ *and* $\mathbf{r_1}|d \in D_{ext}(I)$*.*

We denote by $\mathcal{F}_1(I)$ the set of instances for i_1 compatible with I of type 1, and by $\mathcal{F}_2(I)$ the set of instances for i_1 compatible with I of type 2. We aim to prove the following.

▶ **Lemma 78.** *Let I be an instance for an introduce node i. Then,*

$$\dim(I) = \min\{\min_{I_1 \in \mathcal{F}_1(I)} \{\dim(I_1)\}, \min_{I_2 \in \mathcal{F}_2(I)} \{\dim(I_2) + 1\}\}.$$

Before turning to the proof of Lemma 78, we first show the following technical lemma.

▶ **Lemma 79.** *Let x, s_1, s_2 be three vertices of G_{i_1} and let $\mathbf{r}, \mathbf{t_1}, \mathbf{t_2}$ be three vectors of size k for which there exist $y, z_1, z_2 \notin V(G_i)$ such that $\mathbf{d_{X_i}}(y) = \mathbf{r}$, $\mathbf{d_{X_i}}(z_1) = \mathbf{t_1}$, and $\mathbf{d_{X_i}}(z_2) = \mathbf{t_2}$. Then, the following hold.*

(1) *x is covered by s_1 and $\mathbf{t_1}$ if and only if x is covered by s_1 and $\mathbf{t_1^-}$.*
(2) *x is covered by $(\{\mathbf{t_1}, \mathbf{t_2}\}, d(z_1, z_2))$ if and only if x is covered by $(\{\mathbf{t_1^-}, \mathbf{t_2^-}\}, d(z_1, z_2))$.*
(3) *$\mathbf{r}$ is covered by s_1 and s_2 if and only if $\mathbf{r}^-$ is covered by s_1 and s_2.*
(4) *x is covered by $(\{\mathbf{d_{X_{i_1}}}(v), \mathbf{r_1^-}\}, \mathbf{r_1}_k)$ if and only if x is covered by v and $\mathbf{r_1}$.*

Proof. To prove item (1), it suffices to note that since X_i separates x from z_1 (or $x \in X_i$), $d(z_1, x) = \min_{1 \le j \le k}(\mathbf{t_1} + \mathbf{d_{X_i}}(x))_j$; and for the same reason, $d(z_1, s_1) = \min_{1 \le j \le k}(\mathbf{t_1} + \mathbf{d_{X_i}}(s_1))_j$. But, this is also true of X_{i_1}, and thus, $d(z_1, x) = \min_{1 \le j \le k-1}(\mathbf{t_1} + \mathbf{d_{X_{i_1}}}(x))_j$ and $d(z_1, s_1) = \min_{1 \le j \le k-1}(\mathbf{t_1} + \mathbf{d_{X_{i_1}}}(s_1))_j$. Items (2), (3), and (4) follow from similar arguments. ◀

To prove Lemma 78, we prove the following four lemmas.

▶ **Lemma 80.** *Let I_1 be an instance for i_1 compatible with I of type 1 such that $\dim(I_1) < \infty$, and let S be a minimum-size solution for I_1. Then, S is a solution for I.*

Proof. Let us prove that every condition of Definition 68 is satisfied.

(S1) Let x be a vertex of G_i and assume first that $x \neq v$. Then, since S is a solution for I_1, either x is covered by S, in which case we are done; or (1) x is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I_1)$; or (2) x is covered by an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$. Now, if (1) holds, then, by compatibility, there exists $\mathbf{t} \in D_{ext}(I)$ such that $\mathbf{t}^- = \mathbf{r}$; but then, by Lemma 79(1), x is covered by s and $\mathbf{t}$ (recall indeed that by the definition of $D_{ext}(I)$, there exists $y \notin V(G_i)$ such that $\mathbf{d_{X_i}}(y) = \mathbf{t}$). Suppose next that (2) holds. Then, by compatibility, there exists $(\{\mathbf{t_1}, \mathbf{t_2}\}, d) \in D_{ext/ext}(I)$ such that $\mathbf{t_1^-} = \mathbf{r_1}$ and $\mathbf{t_2^-} = \mathbf{r_2}$; but then, by Lemma 79(2), x is covered by $(\{\mathbf{t_1}, \mathbf{t_2}\}, d)$ (recall indeed that by the definition of $D_{ext/ext}(I)$, there exist $y, z \notin V(G_i)$ such that $\mathbf{d_{X_i}}(y) = \mathbf{t_1}$, $\mathbf{d_{X_i}}(z) = \mathbf{t_2}$, and $d = d(y, z)$).

Assume now that $x = v$. Then, by compatibility, either v is covered by an element of $D_{ext/ext}(I)$, in which case we are done; or (1) v is covered by an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I_1)$; or (2) there exist $\mathbf{r} \in D_{int}(I_1)$ and $\mathbf{t} \in D_{ext}(I)$ such that v is covered by $(\{\mathbf{r}|d, \mathbf{t}\}, \min_{1 \le j \le k}(\mathbf{r}|d + \mathbf{t})_j)$ where $d = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d_{X_{i_1}}}(v))_j$. Now, if (1) holds, then, since S is a solution for I_1, there exist $s_1, s_2 \in S$ such that $\mathbf{d_{X_{i_1}}}(s_1) = \mathbf{r_1}$, $\mathbf{d_{X_{i_1}}}(s_2) = \mathbf{r_2}$, and $d(s_1, s_2) = d$; but then, by Lemma 66(1), v is covered by s_1 and s_2. Now, if (2) holds, then, since S is a solution for I_1, there exists $s \in S$ such that $\mathbf{d_{X_{i_1}}}(s) = \mathbf{r}$; but then, since X_{i_1}

separates s from v (or $s \in X_{i_1}$), $d = d(s, v)$, and thus, by Lemma 66(3), v is covered by s and $\mathbf{t}$.

(S2) Consider a vector $\mathbf{r} \in D_{int}(I)$. Then, by compatibility, $\mathbf{r}_k = \min_{1\leq j\leq k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ and $\mathbf{r}^- \in D_{int}(I_1)$. Now, S is a solution for I_1, and so, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s) = \mathbf{r}^-$; but then, $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{r}$ as X_{i_1} separates s from v (or $s \in X_{i_1}$).

(S3) Consider an element $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$. Then, by compatibility, $\mathbf{r}_k = \min_{1\leq j\leq k-1} (\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ and $\mathbf{t}_k = \min_{1\leq j\leq k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$. Furthermore, either (1) $(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \in D_{int/int}(I_1)$; or (2) $\mathbf{r} = \mathbf{d}_{\mathbf{X_i}}(v)$, $d = \mathbf{t}_k$, and $\mathbf{t}^- \in D_{int}(I_1)$; or (3) $\mathbf{t} = \mathbf{d}_{\mathbf{X_i}}(v)$, $d = \mathbf{r}_k$, and $\mathbf{r}^- \in D_{int}(I_1)$. Now, if (1) holds, then, since S is a solution for I_1, there exist $s_1, s_2 \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s_1) = \mathbf{r}^-$, $\mathbf{d}_{\mathbf{X_{i_1}}}(s_2) = \mathbf{t}^-$, and $d = d(s_1, s_2)$; but then, $\mathbf{d}_{\mathbf{X_i}}(s_1) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X_i}}(s_2) = \mathbf{t}$ as X_{i_1} separates v from both s_1 and s_2 (or s_1 or s_2 belongs to X_{i_1}). Suppose next that (2) holds. Then, since S is a solution for I_1, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s) = \mathbf{t}^-$; but then, $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{t}$ and $d = d(s, v)$ as X_{i_1} separates s from v. Case (3) is handled symmetrically.

(S4) By compatibility, $S_{I_1} = S_I$, and thus, $S \cap X_{i_1} = S \cap (X_i \setminus \{v\}) = S_I = S_{I_1}$. ◀

▶ **Lemma 81.** *Let I_1 be an instance for i_1 compatible with I of type 2 such that $\dim(I_1) < \infty$, and let S be a minimum-size solution for I_1. Then, $S \cup \{v\}$ is a solution for I.*

Proof. Let us prove that the conditions of Definition 68 are satisfied. In the following, we let $S' = S \cup \{v\}$.

(S1) Let x be a vertex of G_i. Since $v \in S'$, we may safely assume that $x \neq v$, that is, $x \in V(G_{i_1})$. Now, S is a solution for I_1, and thus, either x is covered by S, in which case we are done; or (1) x is covered by a vertex $s \in S$ and a vector of $\mathbf{r} \in D_{ext}(I_1)$; or (2) x is covered by an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$.

Suppose first that (1) holds. If $\mathbf{r} = \mathbf{d}_{\mathbf{X_{i_1}}}(v)$, then, by Lemma 66(2), x is covered by $s, v \in S'$. Otherwise, by compatibility, there exists $\mathbf{t} \in D_{ext}(I)$ such that $\mathbf{t}^- = \mathbf{r}$; but then, by Lemma 79(1), x is covered by s and $\mathbf{t}$.

Suppose next that (2) holds. If $\mathbf{r_1} = \mathbf{d}_{\mathbf{X_{i_1}}}(v)$, then, by compatibility, there exists $x \notin V(G_i)$ such that $\mathbf{d}_{\mathbf{X_i}}(x) = \mathbf{r_2}|d$ and $\mathbf{r_2}|d \in D_{ext}(I)$; but then, by Lemma 79(4), x is covered by v and $\mathbf{r_2}|d$. We conclude symmetrically if $\mathbf{r_2} = \mathbf{d}_{\mathbf{X_{i_1}}}(v)$. Thus, by compatibility, we may assume that there exists $(\{\mathbf{t_1}, \mathbf{t_2}\}, d) \in D_{ext/ext}(I)$ such that $\mathbf{t_1^-} = \mathbf{r_1}$ and $\mathbf{t_2^-} = \mathbf{r_2}$; but then, by Lemma 79(2), $(\{\mathbf{t_1}, \mathbf{t_2}\}, d)$ covers x.

(S2) Consider a vector $\mathbf{r} \in D_{int}(I)$ and assume that $\mathbf{r} \neq \mathbf{d}_{\mathbf{X_i}}(v)$ (as $v \in S'$, we are done otherwise). Then, by compatibility, $\mathbf{r}_k = \min_{1\leq j\leq k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ and $\mathbf{r}^- \in D_{int}(I_1)$. Now, S is a solution for I_1, and thus, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s) = \mathbf{r}^-$; but then, $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{r}$ as X_{i_1} separates s from v (or $s \in X_{i_1}$).

(S3) Consider an element $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$. Then, by compatibility, $\mathbf{r}_k = \min_{1\leq j\leq k-1} (\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ and $\mathbf{t}_k = \min_{1\leq j\leq k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$. Furthermore, either (1) $(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \in D_{int/int}(I_1)$; or (2) $\mathbf{r} = \mathbf{d}_{\mathbf{X_i}}(v)$, $d = \mathbf{t}_k$, and $\mathbf{t}^- \in D_{int}(I_1)$; or (3) $\mathbf{t} = \mathbf{d}_{\mathbf{X_i}}(v)$, $d = \mathbf{r}_k$, and $\mathbf{r}^- \in D_{int}(I_1)$. Now, if (1) holds, then, since S is a solution for I_1, there exist $s_1, s_2 \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s_1) = \mathbf{r}^-$, $\mathbf{d}_{\mathbf{X_{i_1}}}(s_2) = \mathbf{t}^-$, and $d(s_1, s_2) = d$; but then, $\mathbf{d}_{\mathbf{X_i}}(s_1) = \mathbf{r}$ and $\mathbf{d}_{\mathbf{X_i}}(s_2) = \mathbf{t}$ as X_{i_1} separates v from both s_1 and s_2 (or s_1 or s_2 belongs to X_{i_1}). Similarly, if (2) holds, then, since S is a solution for I_1, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_{i_1}}}(s) = \mathbf{t}^-$; but then, $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{t}$ as X_{i_1} separates s from v (or $s \in X_{i_1}$). Case 3 is handled symmetrically.

(S4) By compatibility, $S_I = S_{I_1} \cup \{v\}$, and thus, $S \cap X_i = S \cap (X_{i_1} \cup \{v\}) = S_{I_1} \cup \{v\} = S_I$. ◀

As a consequence of Lemmas 80 and 81, the following holds.

▶ **Lemma 82.** *Let I be an instance for an introduce node i. Then,*

$$\dim(I) \leq \min\{\min_{I_1 \in \mathcal{F}_1(I)}\{\dim(I_1)\}, \min_{I_2 \in \mathcal{F}_2(I)}\{\dim(I_2)+1\}\}.$$

▶ **Lemma 83.** *Assume that $\dim(I) < \infty$ and let S be a minimum-size solution for I such that $v \notin S$. Then, there exists $I_1 \in \mathcal{F}_1(I)$ such that S is a solution for I_1.*

Proof. Let I_1 be the instance for i_1 defined as follows.
- $S_{I_1} = S_I$.
- $D_{int}(I_1) = \mathbf{d}_{\mathbf{X_{i_1}}}(S)$.
- $D_{ext}(I_1) = \{\mathbf{r}^- \mid \mathbf{r} \in D_{ext}(I)\}$.
- $D_{int/int}(I_1) = \{(\{\mathbf{d}_{\mathbf{X_{i_1}}}(s_1), \mathbf{d}_{\mathbf{X_{i_1}}}(s_2)\}, d(s_1, s_2)) \mid s_1, s_2 \in S\}$.
- $D_{ext/ext}(I_1) = \{(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \mid (\{\mathbf{r}, \mathbf{t}\}, d) \in D_{ext/ext}(I)\}$.

Let us show that I_1 is compatible with I of type 1 and that S is a solution for I_1.

▷ Claim 84. The constructed instance I_1 is compatible with I of type 1.

Proof. It is clear that condition **(I1)** of Definition 77 holds; let us show that the remaining conditions hold as well.

(I2) Consider a vector $\mathbf{r} \in D_{int}(I)$. Then, since S is a solution for I, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X_i}}(s) = \mathbf{r}$; but then, $\mathbf{r}_k = \min_{1 \leq j \leq k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ as X_{i_1} separates s from v (or $s \in X_{i_1}$), and $\mathbf{r}^- \in D_{int}(I_1)$ by construction.

(I3) Readily follows from the fact that, by construction, $D_{ext}(I_1) = \{\mathbf{r}^- \mid \mathbf{r} \in D_{ext}(I)\}$.

(I4) Consider an element $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$. Then, since S is a solution for I, there exist $s_1, s_2 \in S$ such that $\mathbf{d}_{\mathbf{X_i}}(s_1) = \mathbf{r}$, $\mathbf{d}_{\mathbf{X_i}}(s_2) = \mathbf{t}$, and $d(s_1, s_2) = d$; but then, $\mathbf{r}_k = \min_{1 \leq j \leq k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ and $\mathbf{t}_k = \min_{1 \leq j \leq k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X_{i_1}}}(v))_j$ as X_{i_1} separates v from both s_1 and s_2 (or s_1 or s_2 belongs to X_{i_1}), and $(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \in D_{int/int}(I_1)$ by construction.

(I5) Readily follows from the fact that, by construction, $D_{ext/ext}(I_1) = \{(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \mid (\{\mathbf{r}, \mathbf{t}\}, d) \in D_{ext/ext}(I)\}$.

(I6) Since S is a solution for I and $v \notin S$, either v is covered by an element of $D_{ext/ext}(I)$, in which case we are done; or v is covered by $s_1, s_2 \in S$, in which case $(\{\mathbf{d}_{\mathbf{X_{i_1}}}(s_1), \mathbf{d}_{\mathbf{X_{i_1}}}(s_2)\}, d(s_1, s_2)) \in D_{int/int}(I_1)$ covers v; or v is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I)$, in which case v is covered by $(\{\mathbf{d}_{\mathbf{X_i}}(s), \mathbf{r}\}, \min_{1 \leq j \leq k}(\mathbf{d}_{\mathbf{X_i}}(s) + \mathbf{r})_j)$, where $\mathbf{d}_{\mathbf{X_i}}(s)^- \in D_{int}(I_1)$ by construction. ◁

▷ Claim 85. S is a solution for I_1.

Proof. Let us prove that the conditions of Definition 68 are satisfied.

(S1) Consider a vertex x of G_{i_1}. Then, since $V(G_{i_1}) \subseteq V(G_i)$ and S is a solution for I, either x is covered by S, in which case we are done; or (1) x is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I)$; or (2) x is covered by $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{ext/ext}(I)$. Now, if (1) holds, then, by Lemma 79(1), x remains covered by s and $\mathbf{r}^-$, where $\mathbf{r}^- \in D_{ext}(I_1)$ by construction; and if (2) holds, then, by Lemma 79(2), x remains covered by $(\{\mathbf{r}^-, \mathbf{t}^-\}, d)$, which is an element of $D_{ext/ext}(I_1)$ by construction.

(S2) and **(S3)** readily follow from the fact that, by construction, $D_{int}(I_1) = \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(S)$ and $D_{int/int}(I_1) = \{(\{\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s_1), \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s_2)\}, d(s_1, s_2)) \mid s_1, s_2 \in S\}$, respectively.

(S4) By construction, $S_{I_1} = S_I$, and thus, $S \cap X_{i_1} = S \cap (X_i \setminus \{v\}) = S_I = S_{I_1}$. ◁

The lemma now follows from the above two claims. ◀

▶ **Lemma 86.** *Assume that* $\dim(I) < \infty$ *and let* S *be a minimum-size solution for* I *such that* $v \in S$*. Then, there exists* $I_1 \in \mathcal{F}_2(I)$ *such that* $S \setminus \{v\}$ *is a solution of* I_1*.*

Proof. In the following, we let $S' = S \setminus \{v\}$. Now, let I_1 be the instance for i_1 defined as follows.

- $S_{I_1} = S_I \setminus \{v\}$.
- $D_{int}(I_1) = \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(S')$.
- $D_{ext}(I_1) = \{\mathbf{r}^- \mid \mathbf{r} \in D_{ext}(I)\} \cup \{\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v)\}$.
- $D_{int/int}(I_1) = \{(\{\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s_1), \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(s_2)\}, d(s_1, s_2)) \mid s_1, s_2 \in S'\}$.
- $D_{ext/ext}(I_1) = \{(\{\mathbf{r}, \mathbf{t}\}, d) \mid \exists x, y \notin V(G_i)$ s.t. $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}, \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(y) = \mathbf{t}, d(x, y) = d$, and $(\{\mathbf{r}|d(x, v), \mathbf{t}|d(y, v)\}, d) \in D_{ext/ext}(I)\} \cup \{(\{\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v), \mathbf{r}\}, d) \mid \exists x \notin V(G_{i_1})$ s.t. $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}, d(x, v) = d$, and $\mathbf{r}|d \in D_{ext}(I)\}$.

Let us show that I_1 is compatible with I of type 2, and that S' is a solution for I_1.

▷ Claim 87. The constructed instance I_1 is compatible with I of type 2.

Proof. It is not difficult to see that condition **(I'1)** of Definition 77 holds; let us show that the remaining conditions hold as well.

(I'2) Consider an element $\mathbf{r} \in D_{int}(I) \setminus \{\mathbf{d}_{\mathbf{X}_{\mathbf{i}}}(v)\}$. Then, since S is a solution for I, there exists $s \in S$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}}}(s) = \mathbf{r}$; but then, $\mathbf{r}_k = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v))_j$ as X_{i_1} separates s from v (or $s \in X_{i_1}$), and $\mathbf{r}^- \in D_{int}(I_1)$ by construction.

(I'3) Readily follows from the fact that, by construction, $D_{ext}(I_1) = \{\mathbf{r}^- \mid \mathbf{r} \in D_{ext}(I)\} \cup \{\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v)\}$.

(I'4) Consider an element $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{int/int}(I)$. Then, since S is a solution for I, there exist $s_1, s_2 \in S$ such that $\mathbf{d}_{\mathbf{X}_{\mathbf{i}}}(s_1) = \mathbf{r}$, $\mathbf{d}_{\mathbf{X}_{\mathbf{i}}}(s_2) = \mathbf{t}$, and $d(s_1, s_2) = d$; in particular, $d(s_1, v) = \mathbf{r}_k = \min_{1 \le j \le k-1}(\mathbf{r} + \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v))_j$ and $d(s_2, v) = \mathbf{t}_k = \min_{1 \le j \le k-1}(\mathbf{t} + \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v))_j$, as either $v \in \{s_1, s_2\}$, or X_{i_1} separates v from both s_1 and s_2 (or s_1 or s_2 belongs to X_{i_1}). Now, if, say, $s_1 = v$ (the case where $s_2 = v$ is symmetric), then, by construction, $\mathbf{t}^- \in D_{int}(I_1)$, and thus, the second or third item of **(I'4)** is satisfied. Otherwise, $s_1, s_2 \in S'$, and so, $(\{\mathbf{r}^-, \mathbf{t}^-\}, d) \in D_{int/int}(I_1)$ by construction.

(I'5) Readily follows from the fact that, by construction, $D_{ext/ext}(I_1) = \{(\{\mathbf{r}, \mathbf{t}\}, d) \mid \exists x, y \notin V(G_i)$ s.t. $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}, \mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(y) = \mathbf{t}, d(x, y) = d$, and $(\{\mathbf{r}|d(x, v), \mathbf{t}|d(y, v)\}, d) \in D_{ext/ext}(I)\} \cup \{(\{\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(v), \mathbf{r}\}, d) \mid \exists x \notin V(G_{i_1})$ s.t. $\mathbf{d}_{\mathbf{X}_{\mathbf{i}_1}}(x) = \mathbf{r}, d(x, v) = d$, and $\mathbf{r}|d \in D_{ext}(I)\}$. ◁

▷ Claim 88. S' is a solution for I_1.

Proof. Let us prove that the conditions of Definition 68 are satisfied.

(S1) Consider a vertex x of G_{i_1}. Then, since $V(G_{i_1}) \subseteq V(G_i)$ and S is a solution for I, either (1) x is covered by two vertices $s_1, s_2 \in S$; (2) x is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I)$; or (3) x is covered by an element $(\{\mathbf{r}, \mathbf{t}\}, d) \in D_{ext/ext}(I)$.

Now, if (1) holds, then we may assume that one of s_1 and s_2 is v (otherwise, S' covers x), say $s_1 = v$ without loss of generality; but then, $\mathbf{d_{X_{i_1}}}(v) \in D_{ext}(I_1)$ by construction, and thus, x is covered by $s \in S'$ and $\mathbf{d_{X_{i_1}}}(v) \in D_{ext}(I_1)$ (note indeed that $d(s,v) = \min_{1\leq j\leq k-1}(\mathbf{d_{X_{i_1}}}(s) + \mathbf{d_{X_{i_1}}}(v))_j$ and $d(x,v) = \min_{1\leq j\leq k-1}(\mathbf{d_{X_{i_1}}}(x) + \mathbf{d_{X_{i_1}}}(v))_j$ as X_{i_1} separates v from both s and x).

Suppose next that (2) holds. If $s \neq v$, then, since $\mathbf{r}^- \in D_{ext}(I_1)$ by construction, x remains covered by $s \in S'$ and $\mathbf{r}^- \in D_{ext}(I_1)$ by Lemma 79(1); and if $s = v$, then x is covered by $(\{\mathbf{d_{X_{i_1}}}(v), \mathbf{r}^-\}, \mathbf{r}_k)$ which is an element of $D_{ext/ext}(I_1)$ by construction.

Finally, if (3) holds, then by the definition of $D_{ext/ext}(I)$, there exist $x, y \notin V(G_i)$ such that $\mathbf{d_{X_i}}(x) = \mathbf{r}$, $\mathbf{d_{X_i}}(y) = \mathbf{t}$, and $d(x,y) = d$; but then, by Lemma 66(2), x is covered by $(\{\mathbf{r}^-, \mathbf{t}^-\}, d)$, which is an element of $D_{ext/ext}(I_1)$ by construction.

(S2) and **(S3)** readily follow from the fact that, by construction, $D_{int}(I_1) = \mathbf{d_{X_{i_1}}}(S')$ and $D_{int/int}(I_1) = \{(\{\mathbf{d_{X_{i_1}}}(s_1), \mathbf{d_{X_{i_1}}}(s_2)\}, d(s_1,s_2)) \mid s_1, s_2 \in S'\}$, respectively.

(S4) By construction, $S_{I_1} = S_I \setminus \{v\}$, and so, $S \cap X_{i_1} = S \cap (X_i \setminus \{v\}) = S_I \setminus \{v\} = S_{I_1}$. ◁

The lemma now follows from the above two claims. ◀

As a consequence of Lemmas 83 and 86, the following holds.

▶ **Lemma 89.** *Let I be an instance for an introduce node i. Then,*

$$\dim(I) \geq \min\{\min_{I_1\in\mathcal{F}_1(I)}\{\dim(I_1)\}, \min_{I_2\in\mathcal{F}_2(I)}\{\dim(I_2)+1\}\}.$$

Forget node. Let I be an instance for a forget node i with child i_1, and let $v \in V(G)$ be such that $X_i = X_{i_1} \setminus \{v\}$. Further, let $X_{i_1} = \{v_1, \ldots, v_k\}$, where $v = v_k$.

▶ **Definition 90.** *An instance I_1 for i_1 is compatible with I if the following hold.*

- **(F1)** $S_I = S_{I_1} \setminus \{v\}$.
- **(F2)** *For each $\mathbf{r} \in D_{int}(I)$, there exists $\mathbf{t} \in D_{int}(I_1)$ such that $\mathbf{t}^- = \mathbf{r}$.*
- **(F3)** *For each $\mathbf{r} \in D_{ext}(I_1)$, $\mathbf{r}^- \in D_{ext}(I)$.*
- **(F4)** *For each $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I)$, there exists $(\{\mathbf{t_1}, \mathbf{t_2}\}, d) \in D_{int/int}(I_1)$ such that $\mathbf{t_1}^- = \mathbf{r_1}$ and $\mathbf{t_2}^- = \mathbf{r_2}$.*
- **(F5)** *For each $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$, $(\{\mathbf{r_1}^-, \mathbf{r_2}^-\}, d) \in D_{ext/ext}(I)$.*

We denote by $\mathcal{F}_F(I)$ the set of instances for i_1 compatible with I. We aim to prove the following.

▶ **Lemma 91.** *Let I be an instance for a forget node i. Then,*

$$\dim(I) = \min_{I_1\in\mathcal{F}_F(I)}\{\dim(I_1)\}.$$

Before turning to the proof of Lemma 91, we first prove the following technical lemma, which is the analog of Lemma 79.

▶ **Lemma 92.** *Let x, s_1, s_2 be three vertices of G_i, and let $\mathbf{r}, \mathbf{t_1}, \mathbf{t_2}$ be three vectors of size k for which there exist $y, z_1, z_2 \notin V(G_i)$ such that $\mathbf{d_{X_{i_1}}}(y) = \mathbf{r}$, $\mathbf{d_{X_{i_1}}}(z_1) = \mathbf{t_1}$, and $\mathbf{d_{X_{i_1}}}(z_2) = \mathbf{t_2}$. Then, the following hold.*

(1) *x is covered by s_1 and $\mathbf{t_1}$ if and only if x is covered by s_1 and $\mathbf{t_1^-}$.*

(2) *x is covered by $(\{\mathbf{t_1}, \mathbf{t_2}\}, d(y, z))$ if and only if x is covered by $(\{\mathbf{t_1^-}, \mathbf{t_2^-}\}, d(y, z))$.*

(3) *$\mathbf{r}$ is covered by s_1 and s_2 if and only if $\mathbf{r}^-$ is covered by s_1 and s_2.*

Proof. To prove item (1), it suffices to note that, since X_{i_1} separates x from z_1 (or $x \in X_{i_1}$), $d(z_1, x) = \min_{1 \le j \le k}(\mathbf{t_1} + \mathbf{d_{X_{i_1}}}(x))_j$; and for the same reason, $d(z_1, s_1) = \min_{1 \le j \le k}(\mathbf{t_1} + \mathbf{d_{X_{i_1}}}(s_1))_j$. But, this is also true of X_i, and thus, $d(z_1, x) = \min_{1 \le j \le k-1}(\mathbf{t_1} + \mathbf{d_{X_i}}(x))_j$ and $d(z_1, s_1) = \min_{1 \le j \le k-1}(\mathbf{t_1} + \mathbf{d_{X_i}}(s_1))_j$. Items (2) and (3) follow from similar arguments. ◀

To prove Lemma 91, we prove the following two lemmas.

▶ **Lemma 93.** *Let I_1 be an instance for i_1 compatible with I such that $\dim(I_1) < \infty$, and let S be a minimum-size solution for I_1. Then, S is a solution for I. In particular,*

$$\dim(I) \le \min_{I_1 \in \mathcal{F}_F(I)} \{\dim(I_1)\}.$$

Proof. Let us prove that the conditions of Definition 68 are satisfied.

(S1) Consider a vertex x of G_i. Then, since $V(G_i) = V(G_{i_1})$ and S is a solution for I_1, either x is covered by S, in which case we are done; or (1) x is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I_1)$; or (2) x is covered by an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{ext/ext}(I_1)$. Now, if (1) holds, then, by compatibility, $\mathbf{r}^- \in D_{ext}(I)$; but then, by Lemma 92(1), x is covered by s and $\mathbf{r}^-$. Otherwise, (2) holds, in which case, by compatibility, $(\{\mathbf{r_1^-}, \mathbf{r_2^-}\}, d) \in D_{ext/ext}(I)$; but then, by Lemma 92(2), x is covered by $(\{\mathbf{r_1^-}, \mathbf{r_2^-}\}, d)$.

(S2) Consider a vector $\mathbf{r} \in D_{int}(I)$. Then, by compatibility, there exists $\mathbf{t} \in D_{int}(I_1)$ such that $\mathbf{t}^- = \mathbf{r}$; and since S is a solution for I_1, there then exists $s \in S$ such that $\mathbf{d_{X_{i_1}}}(s) = \mathbf{t}$.

(S3) Consider an element $(\{\mathbf{r_1}, \mathbf{r_2}\}, d) \in D_{int/int}(I)$. Then, by compatibility, there exists $(\{\mathbf{t_1}, \mathbf{t_2}\}, d) \in D_{int/int}(I_1)$ such that $\mathbf{t_1^-} = \mathbf{r_1}$ and $\mathbf{t_2^-} = \mathbf{r_2}$; and since S is a solution for I_1, there then exist $s_1, s_2 \in S$ such that $\mathbf{d_{X_{i_1}}}(s_1) = \mathbf{t_1}$, $\mathbf{d_{X_{i_1}}}(s_2) = \mathbf{t_2}$, and $d(s_1, s_2) = d$.

(S4) By compatibility, $S_I = S_{I_1} \setminus \{v\}$, and thus, $S \cap X_i = S \cap (X_{i_1} \setminus \{v\}) = S_{I_1} \setminus \{v\} = S_I$. ◀

▶ **Lemma 94.** *Assume that $\dim(I) < \infty$ and let S be a minimum-size solution for I. Then, there exists $I_1 \in \mathcal{F}_F(I)$ such that S is a solution for I_1. In particular,*

$$\dim(I) \ge \min_{I_1 \in \mathcal{F}_F(I)} \{\dim(I_1)\}.$$

Proof. Let I_1 be the instance for i_1 defined as follows.

- $S_{I_1} = S \cap X_{i_1}$.
- $D_{int}(I_1) = \mathbf{d_{X_{i_1}}}(S)$.
- $D_{ext}(I_1) = \{\mathbf{r} \mid \exists x \notin V(G_{i_1}) \text{ s.t } \mathbf{d_{X_{i_1}}}(x) = \mathbf{r} \text{ and } \mathbf{r}^- \in D_{ext}(I)\}$.
- $D_{int/int}(I_1) = \{(\{\mathbf{d_{X_{i_1}}}(s_1), \mathbf{d_{X_{i_1}}}(s_2)\}, d(s_1, s_2) \mid s_1, s_2 \in S\}$.

- $D_{ext/ext}(I_1) = \{(\{\mathbf{r},\mathbf{t}\},d) \mid \exists x,y \notin V(G_{i_1}) \text{ s.t } \mathbf{d_{X_{i_1}}}(x) = \mathbf{r}, \mathbf{d_{X_{i_1}}}(y) = \mathbf{t}, d(x,y) = d, \text{ and } (\{\mathbf{r}^-,\mathbf{t}^-\},d) \in D_{ext/ext}(I)\}$.

Let us show that I_1 is compatible with I, and that S is a solution for I_1.

▷ Claim 95. The constructed instance I_1 is compatible with I.

Proof. It is clear that condition **(F1)** of Definition 90 holds; let us show that the remaining conditions hold as well.

(F2) Consider a vector $\mathbf{r} \in D_{int}(I)$. Then, since S is a solution for I, there exists $s \in S$ such that $\mathbf{d_{X_i}}(s) = \mathbf{r}$; but then, $\mathbf{r}|d(s,v) \in D_{int}(I_1)$ by construction.

(F3) readily follows from the fact that $D_{ext}(I_1) = \{\mathbf{r} \mid \exists x \notin V(G_{i_1}) \text{ s.t. } \mathbf{d_{X_{i_1}}}(x) = \mathbf{r} \text{ and } \mathbf{r}^- \in D_{ext}(I)\}$.

(F4) Consider an element $(\{\mathbf{r_1},\mathbf{r_2}\},d) \in D_{int/int}(I)$. Then, since S is a solution for I, there exist $s_1,s_2 \in S$ such that $\mathbf{d_{X_i}}(s_1) = \mathbf{r_1}$, $\mathbf{d_{X_i}}(s_2) = \mathbf{r_2}$, and $d(s_1,s_2) = d$; but then, $(\{\mathbf{r_1}|d(s_1,v),\mathbf{r_2}|d(s_2,v)\},d) \in D_{int/int}(I_1)$ by construction.

(F5) readily follows from the fact that $D_{ext/ext}(I_1) = \{(\{\mathbf{r},\mathbf{t}\},d) \mid \exists x,y \notin V(G_{i_1}) \text{ s.t. } \mathbf{d_{X_{i_1}}}(x) = \mathbf{r}, \mathbf{d_{X_{i_1}}}(y) = \mathbf{t}, d(x,y) = d, \text{ and } (\{\mathbf{r}^-,\mathbf{t}^-\},d) \in D_{ext/ext}(I)\}$. ◁

▷ Claim 96. S is a solution for I_1.

Proof. Let us show that every condition of Definition 68 holds.

(S1) Consider a vertex x of G_{i_1}. Then, since $V(G_i) = V(G_{i_1})$ and S is a solution for I, either x is covered by S, in which case we are done; or (1) x is covered by a vertex $s \in S$ and a vector $\mathbf{r} \in D_{ext}(I)$; (2) or x is covered by an element $(\{\mathbf{r},\mathbf{t}\},d) \in D_{ext/ext}(I)$. Now, suppose that (1) holds and let $y \notin V(G_i)$ be a vertex with distance vector $\mathbf{r}$ to X_i (recall that such a vertex exists by the definition of $D_{ext}(I)$). Then, by Lemma 92(1), x is covered by s and $\mathbf{r}|d(y,v)$; but, by construction, $\mathbf{r}|d(y,v) \in D_{ext}(I_1)$. Suppose next that (2) holds, and let $y,z \notin V(G_i)$ be such that $\mathbf{d_{X_i}}(y) = \mathbf{r}$, $\mathbf{d_{X_i}}(z) = \mathbf{t}$, and $d(y,z) = d$ (recall that such vertices exist by the definition of $D_{ext/ext}(I)$). Then, by Lemma 92(2), x is covered by $(\{\mathbf{r}|d(y,v),\mathbf{t}|d(z,v)\},d(y,z))$, which is an element of $D_{ext/ext}(I_1)$ by construction.

(S2), **(S3)**, and **(S4)** readily follow from the fact that, by construction, $D_{int}(I_1) = \mathbf{d_{X_{i_1}}}(S)$, $D_{int/int}(I_1) = \{(\{\mathbf{d_{X_{i_1}}}(s_1),\mathbf{d_{X_{i_1}}}(s_2)\},d(s_1,s_2) \mid s_1,s_2 \in S\}$, and $S_{I_1} = S \cap X_{i_1}$, respectively. ◁

The lemma now follows from the above two claims. ◀

To complete the proof of Theorem 62, let us now explain how the algorithm proceeds. Given a nice tree decomposition $(T,\mathcal{X})$ of a graph G rooted at node $r \in V(T)$, the algorithm computes the extended geodetic set number for all possible instances in a bottom-up traversal of T. It computes the values for leaf nodes using Lemma 70, for join nodes using Lemma 72, for introduce nodes using Lemma 78, and for forget nodes using Lemma 91. The correctness of this algorithm follows from these lemmas and the following (recall that $\mathrm{gs}(G)$ is the smallest size of a geodetic set for G).

▶ **Lemma 97.** *Let G be a graph and let $(T, \{X_i : i \in V(T)\})$ be a nice tree decomposition of G rooted at node $r \in V(T)$. Then,*

$$\mathrm{gs}(G) = \min_{S_r \subseteq X_r} \dim(X_r, S_r, \emptyset, \emptyset, \emptyset, \emptyset).$$

Proof. Let S be a minimum-size geodetic set of G. Then, by Definition 68, S is a solution for the EGS instance $(X_r, S \cap X_r, \emptyset, \emptyset, \emptyset, \emptyset)$, and so,

$$\min_{S_r \subseteq X_r} \dim(X_r, S_r, \emptyset, \emptyset, \emptyset, \emptyset) \leq \dim(X_r, S \cap X_r, \emptyset, \emptyset, \emptyset, \emptyset) \leq \mathrm{gs}(G).$$

Conversely, let $S' \subseteq X_r$ be a set attaining the minimum above, and let S be a minimum-size solution for the EGS instance $(X_r, S', \emptyset, \emptyset, \emptyset, \emptyset)$. Then, by Definition 68, every vertex of $G_r = G$ is covered by S, and so,

$$\mathrm{gs}(G) \leq \dim(X_r, S', \emptyset, \emptyset, \emptyset, \emptyset) = \min_{S_r \subseteq X_r} \dim(X_r, S_r, \emptyset, \emptyset, \emptyset, \emptyset),$$

which concludes the proof. ◀

Now, let $\alpha(k) = 2^k \cdot 2^{\mathtt{diam}(G)^k} \cdot 2^{\mathtt{diam}(G)^k} \cdot 2^{\mathtt{diam}(G)^{2k+1}} \cdot 2^{\mathtt{diam}(G)^{2k+1}}$. To get the announced complexity, observe first that, at each node $i \in V(T)$, there are at most $\alpha(|X_i|)$ possible instances to consider, where $|X_i| = \mathcal{O}(\mathtt{tw}(G))$; and since T has $\mathcal{O}(\mathtt{tw}(G) \cdot n)$ nodes, there are in total $\mathcal{O}(\alpha(\mathtt{tw}(G)) \cdot \mathtt{tw}(G) \cdot n)$ possible instances. The running time of the algorithm then follows from these facts and the following lemma (note that to avoid repeated computations, we can first compute the distance between every pair of vertices of G in $n^{\mathcal{O}(1)}$ time, as well as all possible distance vectors to a bag from the possible distance vectors to its child/children).

▶ **Lemma 98.** *Let I be an* EGS *instance for a node $i \in V(T)$, and assume that, for every child i_1 of i and every* EGS *instance I_1 for i_1 compatible with I, $\dim(I_1)$ is known. Then, $\dim(I)$ can be computed in time $\alpha(\mathcal{O}(|X_i|)) \cdot n^{\mathcal{O}(1)}$.*

Proof. If i is a leaf node, then $\dim(I)$ can be computed in constant time by Lemma 70. Otherwise, let us prove that one can compute all compatible instances in the child nodes in the announced time (recall that i has at most two child nodes). Given a 6-tuple $I = (X_i, S_I, D_{int}(I), D_{ext}(I), D_{int/int}(I), D_{ext/ext}(I))$, checking whether it is an EGS instance can be done in $\mathcal{O}(|I|) \cdot n^{\mathcal{O}(1)}$ time; and the number of such 6-tuples is bounded by $\alpha(|X_i|)$. It is also not difficult to see that checking for compatibility can, in each case, be done in $\mathcal{O}(|I|) \cdot n^{\mathcal{O}(1)}$ time. Now, note that, by Definition 68, $|I| = \mathtt{diam}(G)^{\mathcal{O}(|X_i|)}$, and thus, computing all compatible instances can indeed be done in $\alpha(\mathcal{O}(|X_i|)) \cdot n^{\mathcal{O}(1)}$ time. Then, since computing the minimum using the formulas of Lemmas 72, 78, and 91 can be done in $\alpha(\mathcal{O}(|X_i|))$ time, the lemma follows. ◀

8.3 (Kernelization) Algorithm for Strong Metric Dimension

We prove the following theorem.

▶ **Theorem 99.** *Strong Metric Dimension admits*
- *an* FPT *algorithm running in time $2^{2^{\mathcal{O}(\mathtt{vc})}} \cdot n^{\mathcal{O}(1)}$, and*
- *a kernelization algorithm that outputs a kernel with $2^{\mathcal{O}(\mathtt{vc})}$ vertices.*

Proof. Given a graph G, let $X \subseteq V(G)$ be a minimum vertex cover of G. If such a vertex cover is not given, then we can find a 2-factor approximate vertex cover in polynomial time. Let $I := V(G) \setminus X$. By the definition of a vertex cover, the vertices of I are pairwise non-adjacent. The kernelization algorithm exhaustively applies the following reduction rule.

Reduction Rule 1. *If there exist three vertices $u, v, x \in I$ such that u, v, x are false twins, then delete x and decrease k by one.*

Since u, v, x are false twins, $N(u) = N(v) = N(x)$. This implies that, for any vertex $w \in V(G) \setminus \{u, v, x\}$, $d(w, v) = d(w, u) = d(w, x)$. In other words, for any $w \neq v$, any shortest path from u to w does not contain v. Hence, any strong resolving set that excludes at least two vertices in $\{u, v, x\}$ cannot resolve all three pairs $\langle u, v\rangle$, $\langle u, x\rangle$, and $\langle v, x\rangle$. Hence, we can assume, without loss of generality, that any resolving set contains both u and x.

Any pair of vertices in $V(G) \setminus \{u, x\}$ that is strongly resolved by x is also resolved by u. In other words, if S is a strong resolving set of G, then $S \setminus \{x\}$ is a strong resolving set of $G - \{x\}$. This implies the correctness of the forward direction. The correctness of the reverse direction trivially follows from the fact that we can add x into a strong resolving set of $G - \{x\}$ to obtain a resolving set of G.

Consider an instance on which the reduction rule is not applicable. If the budget is negative, then the algorithm returns a trivial No-instance of constant size. Otherwise, for any $Y \subseteq X$, there are at most two vertices $u, v \in I$ such that $N(u) = N(v) = Y$. This implies that the number of vertices in the reduced instance is at most $|X| + 2 \cdot 2^{|X|} = 2^{\texttt{vc}+1} + \texttt{vc}$. The second part of the statement is an immediate consequence of applying a brute-force algorithm on the reduced instance. ◀

9 Conclusion

We have shown (under the ETH) that three natural metric-based graph problems, Metric Dimension, Geodetic Set, and Strong Metric Dimension, exhibit tight (double-) exponential running times for the standard structural parameterizations by treewidth and vertex cover number. This includes tight double-exponential running times for treewidth plus diameter (Metric Dimension and Geodetic Set) and for vertex cover (Strong Metric Dimension).

Such tight double-exponential running times for FPT structural paramaterizations of graph problems had previously been observed only for counting problems and problems complete for classes above NP. Thus, surprisingly, our results show that some natural problems can be in NP and still exhibit such a behavior.

It would be interesting to see whether this phenomenon holds for other graph problems in NP, and for other structural parameterizations. Perhaps one can determine certain properties shared by these metric-based graph problems, that imply such running times, with the goal of generalizing our approach to a broader class of problems. In particular, concerning the general versatile technique that we designed to obtain the double-exponential lower bounds, it would be intriguing to see for which other problems in NP our technique works.

In fact, after this paper appeared online, our technique was successfully applied to an NP-complete problem in machine learning [19] (for vc) as well as NP-complete identification problems [17] (for tw).

References

1 A. Achilleos, M. Lampis, and V. Mitsou. Parameterized modal satisfiability. *Algorithmica*, 64(1):38–55, 2012.

2 J. Ahn, L. Jaffke, O. Kwon, and P. T. Lima. Well-partitioned chordal graphs. *Discrete Mathematics*, 345(10):112985, 2022.

3 S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. URL: http://www.cs.princeton.edu/theory/complexity/.

4 M. Atici. Computational complexity of geodetic set. *International journal of computer mathematics*, 79(5):587–591, 2002.

5 L. Babai. On the complexity of canonical labelling of strongly regular graphs. *SIAM J. Comput.*, 9(1):212–216, 1980.

6 F. Barbero, L. Isenmann, and J. Thiebaut. On the distance identifying set meta-problem and applications to the complexity of identifying problems on graphs. *Algorithmica*, 82(8):2243–2266, 2020.

7 Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffman, M. Mihalák, and L. S. Ram. Network discovery and verification. *IEEE J. Sel. Area Comm.*, 24(12):2168–2181, 2006.

8 R. Belmonte, F. V. Fomin, P. A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM J. Discrete Math.*, 31(2):1217–1243, 2017.

9 Y. Ben-Haim, S. Gravier, A. Lobstein, and J. Moncel. Adaptive identification in graphs. *J. Comb. Theory, Ser. A*, 115(7):1114–1126, 2008.

10 J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and S. Pérennes. Sequential metric dimension. *Algorithmica*, 82(10):2867–2901, 2020.

11 B. Bergougnoux, O. Defrain, and F. Mc Inerney. Enumerating minimal solution sets for metric graph problems. In *Proc. of the 50th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2024)*, Lecture Notes in Computer Science. Springer, 2024.

12 E. Bonnet and N. Purohit. Metric dimension parameterized by treewidth. *Algorithmica*, 83:2606–2633, 2021.

13 N. Bousquet, Q. Deschamps, and A. Parreau. Metric dimension parameterized by treewidth in chordal graphs. In *Proc. of the 49th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2023)*, volume 14093 of *Lecture Notes in Computer Science*, pages 130–142. Springer, 2023.

14 L. R. Bueno, L. D. Penso, F. Protti, V. R. Ramos, D. Rautenbach, and U. S. Souza. On the hardness of finding the geodetic number of a subcubic graph. *Inf. Process. Lett.*, 135:22–27, 2018.

15 D. Chakraborty, S. Das, F. Foucaud, H. Gahlawat, D. Lajou, and B. Roy. Algorithms and complexity for geodetic sets on planar and chordal graphs. In *31st International Symposium on Algorithms and Computation (ISAAC 2020)*, volume 181 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:15. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

16 D. Chakraborty, F. Foucaud, H. Gahlawat, S. K. Ghosh, and B. Roy. Hardness and approximation for the geodetic set problem in some graph classes. In *Proc. of the 6th International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2020)*, volume 12016 of *Lecture Notes in Computer Science*, pages 102–115, Cham, 2020. Springer.

17 D. Chakraborty, F. Foucaud, D. Majumdar, and P. Tale. Tight (double) exponential bounds for identification problems: Locating-dominating set and test cover, 2024. URL: `http://arxiv.org/abs/2402.08346`, `arXiv:2402.08346`.

18 D. Chakraborty, H. Gahlawat, and B. Roy. Algorithms and complexity for geodetic sets on partial grids. *Theoretical Computer Science*, 979:114217, 2023.

19 J. Chalopin, V. Chepoi, F. Mc Inerney, and S. Ratel. Non-clashing teaching maps for balls in graphs. *Arxiv:2309.02876*, 2023.

20 G. Chartrand, F. Harary, and P. Zhang. On the geodetic number of a graph. *Networks*, 39(1):1–6, 2002.

21 H. Chen. Quantified constraint satisfaction and bounded treewidth. In *Proc. of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, pages 161–165. IOS Press, 2004.

22 B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

23 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

24 M. Cygan, D. Marx, M. Pilipczuk, and M. Pilipczuk. Hitting forbidden subgraphs in graphs of bounded treewidth. *Inf. Comput.*, 256(C):62–82, 2017.

25 M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 150–159, 2011.

26 M. Cygan, M. Pilipczuk, and M. Pilipczuk. Known algorithms for edge clique cover are probably optimal. *SIAM J. Comput.*, 45(1):67–83, 2016.

27 B. DasGupta and N. Mobasheri. On optimal approximability results for computing the strong metric dimension. *Discrete Applied Math.*, 221:18–24, 2017.

28 T. Davot, L. Isenmann, and J. Thiebaut. On the approximation hardness of geodetic set and its variants. In *Proc. of the 27th International Computing and Combinatorics Conference, COCOON 2021*, volume 13025 of *Lecture Notes in Computer Science*, pages 76–88. Springer, 2021.

29 J. Díaz, O. Pottonen, M. J. Serna, and E. J. van Leeuwen. Complexity of metric dimension on planar graphs. *J. Comput. Syst. Sci.*, 83(1):132–158, 2017.

30 R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

31 M. C. Dourado, F. Protti, D. Rautenbach, and J. L. Szwarcfiter. On the complexity of the geodetic and convexity numbers of a graph. In *Proc. of the International Conference on Discrete Mathematics (ICDM)*, volume 7 of *RMS Lecture Notes Series*, pages 101–108. Ramanujan Mathematical Society, 2008.

32 M. C. Dourado, F. Protti, D. Rautenbach, and J. L. Szwarcfiter. Some remarks on the geodetic number of a graph. *Discrete Mathematics*, 310(4):832–837, 2010.

33 A. L. Douthat and M. C. Kong. Computing geodetic bases of chordal and split graph. *Journal of Combinatorial Mathematics and Combinatorial Computing*, pages 67–77, 1996.

34 T. Ekim, A. Erey, P. Heggernes, P. van't Hof, and D. Meister. Computing minimum geodetic sets of proper interval graphs. In *Proc. of the 10th Latin American Symposium on Theoretical Informatics (LATIN 2012)*, volume 7256 of *Lecture Notes in Computer Science*, pages 279–290. Springer, 2012.

35 D. Eppstein. Metric dimension parameterized by max leaf number. *Journal of Graph Algorithms and Applications*, 19(1):313–323, 2015.

36 L. Epstein, A. Levin, and G. J. Woeginger. The (weighted) metric dimension of graphs: Hard and easy cases. *Algorithmica*, 72(4):1130–1171, 2015.

37 A. Estrada-Moreno, J. A. Rodriguez-Velázquez, and I. G. Yero. The k-metric dimension of a graph. *Applied Mathematics and Information Sciences*, 9(6):2829–2840, 2015.

38 M. Farber and R. E. Jamison. Convexity in graphs and hypergraphs. *SIAM Journal on Algebraic Discrete Methods*, 7(3):433–444, 1986.

39 H. Fernau, P. Heggernes, P. van 't Hof, D. Meister, and R. Saei. Computing the metric dimension for chain graphs. *Inf. Process. Lett.*, 115(9):671–676, 2015.

40 J. K. Fichte, M. Hecher, M. Morak, P. Thier, and S. Woltran. Solving projected model counting by utilizing treewidth and its limits. *Artif. Intell.*, 314:103810, 2023.

41 J. K. Fichte, M. Hecher, M. Morak, and S. Woltran. Exploiting treewidth for projected model counting and its limits. In *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Proc.*, volume 10929 of *Lecture Notes in Computer Science*, pages 165–184. Springer, 2018.

42 J. K. Fichte, M. Hecher, and A. Pfandler. Lower bounds for QBFs of bounded treewidth. In *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 410–424. ACM, 2020.

43 M. J. Fischer and M. O. Rabin. Super-exponential complexity of presburger arithmetic. In *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 122–135. Springer, 1998.

44 J. Focke, F. Frei, S. Li, D. Marx, P. Schepper, R. Sharma, and K. Wegrzycki. Hitting meets packing: How hard can it be? *CoRR*, abs/2402.14927, 2024. URL: `https://doi.org/10.48550/arXiv.2402.14927`.

45 F. V. Fomin, P. A. Golovach, D. Lokshtanov, S. Saurabh, and M. Zehavi. Clique-width III: hamiltonian cycle and the odd case of graph coloring. *ACM Trans. Algorithms*, 15(1):9:1–9:27, 2019.

46 F. Foucaud, E. Galby, L. Khazaliya, S. Li, F. Mc Inerney, R. Sharma, and P. Tale. Problems in NP can admit double-exponential lower bounds when parameterized by treewidth or vertex cover. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

47 F. Foucaud, G. B. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. II. Algorithms and complexity. *Algorithmica*, 78(3):914–944, 2017.

48 M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48(6):1184–1206, 2001.

49 R. M. Frongillo, J. Geneson, M. E. Lladser, R. C. Tillquist, and E. Yi. Truncated metric dimension for finite graphs. *Discrete Applied Mathematics*, 320:150–169, 2022.

50 E. Galby, L. Khazaliya, F. Mc Inerney, R. Sharma, and P. Tale. Metric dimension parameterized by feedback vertex set and other structural parameters. *SIAM J. Discrete Math.*, 37(4):2241–2264, 2023.

51 M. R. Garey and D. S. Johnson. *Computers and Intractability - A guide to NP-completeness.* W.H. Freeman and Company, 1979.

52 T. Gima, T. Hanaka, M. Kiyomi, Y. Kobayashi, and Y. Otachi. Exploring the gap between treedepth and vertex cover through vertex integrity. *Theoretical Computer Science*, 918:60–76, 2022.

53 T. Hanaka, H. Köhler, and M. Lampis. Core stability in additively separable hedonic games of low treewidth, 2024. URL: `http://arxiv.org/abs/2402.10815`, `arXiv:2402.10815`.

54 F. Harary, E. Loukakis, and C. Tsouros. The geodetic number of a graph. *Mathematical and Computer Modelling*, 17(11):89–95, 1993.

55 F. Harary and R. A. Melter. On the metric dimension of a graph. *Ars Combinatoria*, 2:191–195, 1976.

56 S. Hartung and A. Nichterlein. On the parameterized and approximation hardness of metric dimension. In *Proc. of the 28th Conference on Computational Complexity, CCC 2013*, pages 266–276. IEEE Computer Society, 2013.

57 S. Hoffmann, A. Elterman, and E. Wanke. A linear time algorithm for metric dimension of cactus block graphs. *Theoretical Computer Science*, 630:43–62, 2016.

58 S. Hoffmann and E. Wanke. Metric dimension for gabriel unit disk graphs is NP-complete. In *8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS 2012)*, pages 90–92, Berlin, Heidelberg, 2013. Springer.

59 A. Ian. *Combinatorics of Finite Sets.* Oxford University Press, 1987.

60 L. Jaffke, O j. Kwon, T. J. F. Strømme, and J. A. Telle. Mim-width III. Graph powers and generalized distance domination problems. *Theoretical Computer Science*, 796:216–236, 2019.

61 K. Jansen, KM. Klein, and A. Lassota. The double exponential runtime is tight for 2-stage stochastic ILPs. *Math. Program.*, 197:1145–1172, 2023.

62 M. A. Johnson. Structure-activity maps for visualizing the graph variables arising in drug design. *J. Biopharm. Statist.*, 3:203–236, 1993.

63 M. M. Kanté and L. Nourine. Polynomial time algorithms for computing a minimum hull set in distance-hereditary and chordal graphs. *SIAM J. Discrete Math.*, 30(1):311–326, 2016.

64 M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Trans. Information Theory*, 44(2):599–611, 1998.

65 I. Katsikarelis, M. Lampis, and V. Th. Paschos. Structurally parameterized d-scattered set. *Discrete Applied Mathematics*, 308:168–186, 2022.

66 L. Kellerhals and T. Koana. Parameterized complexity of geodetic set. *Journal of Graph Algorithms and Applications*, 26(4):401–419, 2022.

67 J. M. Kleinberg and E. Tardos. *Algorithm design.* Addison-Wesley, 2006.

68 T. Kloks. *Treewidth, Computations and Approximations.* Springer, 1994.

69 D. Knop, M. Pilipczuk, and M. Wrochna. Tight complexity lower bounds for integer linear programming with few constraints. *ACM Trans. Comput. Theory*, 12(3):19:1–19:19, 2020.

70 T. Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 184–192, 2022.

71 L. Kowalik, A. Lassota, K. Majewski, M. Pilipczuk, and M. Sokołowski. Detecting points in integer cones of polytopes is double-exponentially hard. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 279–285, 2024.

72 M. Künnemann, F. Mazowiecki, L. Schütze, H. Sinclair-Banks, and K. Węgrzycki. Coverability in VASS Revisited: Improving Rackoff's Bound to Obtain Conditional Optimality. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

73 D. Kuziak, M. L. Puertas, J. A. Rodríguez-Velázquez, and I. G. Yero. Strong resolving graphs: The realization and the characterization problems. *Discrete Applied Math.*, 236:270–287, 2018.

74 M. Lampis, N. Melissinos, and M. Vasilakis. Parameterized max min feedback vertex set. In *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023*, volume 272 of *LIPIcs*, pages 62:1–62:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

75 M. Lampis, S. Mengel, and V. Mitsou. QBF as an alternative to courcelle's theorem. In *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018*, volume 10929 of *Lecture Notes in Computer Science*, pages 235–252. Springer, 2018.

76 M. Lampis and V. Mitsou. Treewidth with a quantifier alternation revisited. In *12th International Symposium on Parameterized and Exact Computation (IPEC 2017)*, volume 89 of *LIPIcs*, pages 26:1–26:12, 2017.

77 S. Li and M. Pilipczuk. Hardness of metric dimension in graphs of constant treewidth. *Algorithmica*, 84(11):3110–3155, 2022.

78 D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018.

79 D. Lokshtanov, S. Saurabh, S. Suri, and J. Xue. An ETH-tight algorithm for multi-team formation. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021*, volume 213 of *LIPIcs*, pages 28:1–28:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

80 D. Marx and V. Mitsou. Double-exponential and triple-exponential bounds for choosability problems parameterized by treewidth. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 28:1–28:15, 2016.

81 C. Mathieu and H. Zhou. A simple algorithm for graph reconstruction. In *29th Annual European Symposium on Algorithms, ESA 2021*, volume 204 of *LIPIcs*, pages 68:1–68:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

82 R. A. Melter and I. Tomescu. Metric bases in digital geometry. *Comput. Vision Graphics Image Process.*, 25:113–121, 1984.

83 M. Mezzini. Polynomial time algorithm for computing a minimum geodetic set in outerplanar graphs. *Theoretical Computer Science*, 745:63–74, 2018.

84 M. Moscarini. Computing a metric basis of a bipartite distance-hereditary graph. *Theoretical Computer Science*, 900:20–24, 2022.

85 O. R. Oellermann and J. Peters-Fransen. The strong metric dimension of graphs and digraphs. *Discrete Applied Mathematics*, 155(3):356–364, 2007.

86 I. M. Pelayo. *Geodesic Convexity in Graphs.* Springer, 2013.

87 M. Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Proc.*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011.

88 M. Pilipczuk and M. Sorge. A double exponential lower bound for the distinct vectors problem. *Discret. Math. Theor. Comput. Sci.*, 22(4), 2020.

89 A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. In *Automata, Languages and Programming, 16th International Colloquium, ICALP89, Proc.*, volume 372 of *Lecture Notes in Computer Science*, pages 652–671. Springer, 1989.

90 J. Rintanen. Complexity of planning with partial observability. In *Proc. of the 14th International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 345–354. AAAI, 2004.

91 A. Sebő and E. Tannier. On metric generators of graphs. *Mathematics of Operations Research*, 29(2):383–393, 2004.

92 P. J. Slater. Leaves of trees. In *Proc. of the 6th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 549–559. Congressus Numerantium, No. XIV. Utilitas Mathematica, 1975.

93 P. J. Slater. Domination and location in acyclic graphs. *Networks*, 17(1):55–64, 1987.

94 P. Tale. Double exponential lower bound for telephone broadcast, 2024. URL: `http://arxiv.org/abs/2403.03501`, `arXiv:2403.03501`.

95 R. C. Tillquist and M. E. Lladser. Low-dimensional representation of genomic sequences. *Journal of Mathematical Biology*, 79:1–29, 2019.