Explaining Decisions in ML Models: a Parameterized Complexity Analysis (Full Paper)

Sebastian Ordyniak[1] , **Giacomo Paesani**[2] , **Mateusz Rychlicki**[1] , **Stefan Szeider**[3]
[1]University of Leeds, Leeds, UK
[2]Sapienza University of Rome, Rome, Italy
[3]TU Wien, Vienna, Austria
{sordyniak,giacomopaesani,mkrychlicki}@gmail.com, sz@ac.tuwien.ac.at

Abstract

This paper presents a comprehensive theoretical investigation into the parameterized complexity of explanation problems in various machine learning (ML) models. Contrary to the prevalent black-box perception, our study focuses on models with transparent internal mechanisms. We address two principal types of explanation problems: abductive and contrastive, both in their local and global variants. Our analysis encompasses diverse ML models, including Decision Trees, Decision Sets, Decision Lists, Ordered Binary Decision Diagrams, Random Forests, and Boolean Circuits, and ensembles thereof, each offering unique explanatory challenges. This research fills a significant gap in explainable AI (XAI) by providing a foundational understanding of the complexities of generating explanations for these models. This work provides insights vital for further research in the domain of XAI, contributing to the broader discourse on the necessity of transparency and accountability in AI systems.

1 Introduction

As machine learning (ML) models increasingly permeate essential domains, understanding their decision-making mechanisms has become central. This paper delves into the field of explainable AI (XAI) by examining the parameterized complexity of explanation problems in various ML models. We focus on models with accessible internal mechanisms, shifting away from the traditional black-box paradigm. Our motivation is rooted in establishing a comprehensive theoretical framework that illuminates the complexity of generating explanations for these models, a task becoming increasingly relevant in light of recent regulatory guidelines that emphasize the importance of transparent and explainable AI (Commission 2020; OECD 2023).

The need for transparency and accountability in automated decision-making drives the imperative for explainability in AI systems, especially in high-risk sectors. ML models, while powerful, must be demystified to gain trust and comply with ethical and regulatory standards. Formal explanations serve this purpose, providing a structured means to interpret model decisions (Marques-Silva 2023; Guidotti et al. 2019; Carvalho, Pereira, and Cardoso 2019).

Our exploration focuses on two types of explanation problems, abductive and contrastive, in local and global contexts (Marques-Silva 2023). *Abductive explanations* (Ignatiev, Narodytska, and Marques-Silva 2019), corresponding to prime-implicant explanations (Shih, Choi, and Darwiche 2018) and sufficient reason explanations (Darwiche and Ji 2022), clarify specific decision-making instances, while *contrastive explanations* (Miller 2019; Ignatiev et al. 2020), corresponding to necessary reason explanations (Darwiche and Ji 2022), make explicit the reasons behind the non-selection of alternatives. The study of contrastive explanations goes back to the Lipton's work in 1990. Conversely, *global explanations* (Ribeiro, Singh, and Guestrin 2016; Ignatiev, Narodytska, and Marques-Silva 2019) aim to unravel models' decision patterns across various inputs. This bifurcated approach enables a comprehensive understanding of model behavior, aligning with the recent emphasis on interpretable ML (Lisboa et al. 2023).

In contrast to a recent study by Ordyniak, Paesani, and Szeider (2023), who consider the parameterized complexity of finding explanations based on samples classified by a black-box ML model, we focus on the setting where the model together with its inner workings is available as an input for computing explanations. This perspective, initiated by Barceló et al. (2020), is particularly appealing, as it lets us quantify the explainability of various model types based on the computational complexity of the corresponding explanation problems.

Challenging the notion of inherent opacity in ML models, our study includes *Decision Trees* (DTs), *Decision Sets* (DSs), *Decision Lists* (DLs), and *Ordered Binary Decision Diagrams* (OBDDs). Whereas DTs, DSs, and DLs are classical ML models, OBDDs can be used to represent the decision, functions of naive Bayes classifiers (Chan and Darwiche 2003). We also consider *ensembles* of all the above ML models; where an ensemble classifies an example by taking the majority classification over its elements. For instance, *Random Forests* (RFs) are ensembles of DTs.

Each model presents distinct features affecting explanation generation. For example, the transparent structure of DTs and RFs facilitates rule extraction, as opposed to the complex architectures of *Neural Networks* (NNs) (Ribeiro, Singh, and Guestrin 2016; Lipton 2018).

Contribution This paper fills a crucial gap in XAI research by analyzing the complexity of generating explanations across different models. Prior research has often centred on practical explainability approaches, but a theoretical under-

standing still needs to be developed (Holzinger et al. 2020; Molnar 2023). Our study is aligned with the increasing call for theoretical rigor in AI (Commission 2019). By dissecting the parameterized complexity of these explanation problems, we lay the groundwork for future research and algorithm development, ultimately contributing to more efficient explanation methods in AI.

Since most of the considered explanation problems are NP-hard, we use the paradigm of fixed-parameter tractability (FPT), which involves identifying specific parameters of the problem (e.g., explanation size, number of terms/rules, size/height of a DT, width of a BDD) and proving that the problem is fixed-parameter tractable concerning these parameters. By focusing on these parameters, the complexity of the problem is confined, making it more manageable and often solvable in uniform polynomial time for fixed values of the parameters. A significant part of our positive results are based on reducing various model types to *Boolean circuits (*BC*s)*. This reduction is crucial for the uniform treatment of several model types as it allows the application of known algorithmic results and techniques from the Boolean circuits domain to the studied models. It simplifies the problems and brings them into a well-understood theoretical framework. For ensembles, we consider Boolean circuits with majority gates. In turn, we obtain the fixed-parameter tractability of problems on Boolean circuits via results on Monadic Second Order (MSO). We use extended MSO (Bergougnoux, Dreier, and Jaffke 2023) to handle majority gates, which allows us to obtain efficient algorithmic solutions, particularly useful for handling complex structures.

Overall, the approach in the manuscript is characterized by a mix of theoretical computer science techniques, including parameterization, reduction to well-known problems, and the development of specialized algorithms that exploit the structural properties of the models under consideration. This combination enables the manuscript to effectively address the challenge of finding tractable solutions to explanation problems in various machine learning models.

For some of the problems, we develop entirely new customized algorithms. We complement the algorithmic results with hardness results to get a complete picture of the tractability landscape for all possible combinations of the considered parameters (an overview of our results are provided in Tables 2, 3, 4).

In summary, our research marks a significant advancement in the theoretical understanding of explainability in AI. By offering a detailed complexity analysis for various ML models, this work enriches academic discourse and responds to the growing practical and regulatory demand for transparent, interpretable, and trustworthy AI systems.

A full version of the paper can be found on ArXiv (Ordyniak et al. 2024).

2 Preliminaries

For a positive integer i, we denote by $[i]$ the set of integers $\{1, \ldots, i\}$.

Parameterized Complexity (PC). We outline some basic concepts refer to the textbook by Downey and Fellows (2013) for an in-depth treatment. An instance of a parameterized problem Q is a pair (x, k) where x is the main part and k (usually an non-negative integer) is the parameter. Q is *fixed-parameter tractable (FPT)* if it can be solved in time $f(k)n^c$ where n is the input size of x, c is a constant independent of k, and f is a computable function. If a problem has more then one parameters, then the parameters can be combined to a single one by addition. FPT denotes the class of all fixed-parameter tractable decision problems. XP denotes the class of all parameterized decision problems solvable in time $n^{f(k)}$ where f is again a computable function. An *fpt-reduction* from one parameterized decision problem Q to another Q' is an fpt-computable reduction that reduces of Q to instances of Q' such that yes-instances are mapped to yes-instances and no-instances are mapped to no-instances. The parameterized complexity classes W[i] are defined as the closure of certain weighted circuit satisfaction problems under fpt-reductions. Denoting by P the class of all parameterized decision problems solvable in polynomial time, and by paraNP the class of parameterized decision problems that are in NP and NP-hard for at least one instantiation of the parameter with a constant, we have $\mathsf{P} \subseteq \mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{XP} \cap \mathsf{paraNP} \subseteq \mathsf{paraNP}$, where all inclusions are believed to be strict. If a parameterized problem is W[i]-$hard$ under fpt-reductions (W[i]-h, for short) then it is unlikely to be FPT. co-C denotes the complexity class containing all problems from C with yes-instances replaced by no-instances and no-instances replaced by yes-instances.

Graphs, Rankwidth, Treewidth and Pathwidth. We mostly use standard notation for graphs as can be found, e.g., in (Diestel 2000). Let $G = (V, E)$ be a directed or undirected graph. For a vertex subset $V' \subseteq V$, we denote by $G[V']$ the graph induced by the vertices in V' and by $G \setminus V'$ the graph $G[V \setminus V']$. If G is directed, we denote by $N_G^-(v)$ ($N_G^+(v)$) the set of all incoming (outgoing) neighbors of the vertex $v \in V$.

Let $G = (V, E)$ be a directed graph. A *tree decomposition* of G is a pair $\mathcal{T} = (T, \lambda)$ with T being a tree and $\lambda : V(T) \to V(G)$ such that: (1) for every vertex $v \in V$ the set $\{\, t \in V(T) \mid v \in \lambda(t) \,\}$ is forms a non-empty subtree of T and (2) for every arc $e = (u, v) \in E$, there is a node $t \in V(T)$ with $u, v \in \lambda(t)$. The *width* of $\mathcal{T}$ is equal to $\max_{t \in V(T)} |\lambda(t)| - 1$ and the *treewidth* of G is the minimum width over all tree decompositions of G. $\mathcal{T}$ is called a *path decomposition* if T is a path and the *pathwidth* of G is the minimum width over all path decompositions of G. We will need the following well-known properties of pathwidth, treewidth, and rankwidth.

Lemma 1 ((Corneil and Rotics 2001; Oum and Seymour 2006))**.** *Let $G = (V, E)$ be a directed graph and $X \subseteq V$. The treewidth of G is at most $|X|$ plus the treewidth of $G - X$. Furthermore, if G has rankwidth r, pathwidth p and treewidth t, then $r \leq 3 \cdot 2^{t-1} \leq 3 \cdot 2^{p-1}$.*

Examples and Models Let F be a set of binary features. An *example* $e : F \to \{0, 1\}$ over F is a $\{0, 1\}$-assignment of the features in F. An example is a *partial example (assignment)* over F if it is an example over some subset F' of F. We denote by $E(F)$ the set of all possible examples over F. A

(binary classification) model $M : E(F) \to \{0,1\}$ is a specific representation of a Boolean function over $E(F)$. We denote by $F(M)$ the set of features considered by M, i.e., $F(M) = F$. We say that an example e is a 0-example or negative example (1-example or positive example) w.r.t. the model M if $M(e) = 0$ ($M(e) = 1$). For convenience, we restrict our setting to the classification into two classes. We note however that all our hardness results easily carry over to the classification into any (in)finite set of classes. The same applies to our algorithmic results for non-ensemble models since one can easily reduce to the case with two classes by renaming the class of interest for the particular explanation problem to 1 and all other classes to 0. We leave it open whether the same holds for our algorithmic results for ensemble models.

Decision Trees. A *decision tree* (DT) $\mathcal{T}$ is a pair (T, λ) such that T is a rooted binary tree and $\lambda : V(T) \to F \cup \{0,1\}$ is a function that assigns a feature in F to every inner node of T and either 0 or 1 to every leaf node of T. Every inner node of T has exactly 2 children, one left child (or 0-child) and one right-child (or 1-child). The classification function $\mathcal{T} : E(F) \to \{0,1\}$ of a DT is defined as follows for an example $e \in E(F)$. Starting at the root of T one does the following at every inner node t of T. If $e(\lambda(t)) = 0$ one continues with the 0-child of t and if $e(\lambda(t)) = 1$ one continues with the 1-child of t until one eventually ends up at a leaf node l at which e is classified as $\lambda(l)$. For every node t of T, we denote by $\alpha^t_{\mathcal{T}}$ the partial assignment of F defined by the path from the root of T to t in T, i.e., for a feature f, we set $\alpha^t_{\mathcal{T}}(f)$ to 0 (1) if and only if the path from the root of T to t contains an inner node t' with $\lambda(t') = f$ together with its 0-child (1-child). We denote by $L(\mathcal{T})$ the set of leaves of T and we set $L_b(\mathcal{T}) = \{\, l \in L(\mathcal{T}) \mid \lambda(l) = b \,\}$ for every $b \in \{0,1\}$. Moreover, we denote by $\|\mathcal{T}\|$ ($h(\mathcal{T})$) the size (height) of a DT, which is equal to the number of leaves of T (the length of a longest root-to-leaf path in T). Finally, we let $\mathsf{MNL}(\mathcal{T}) = \min\{|L_0|, |L_1|\}$.

Decision Sets. A *term* t over F is a set of *literals* with each literal being of the form $(f = z)$ where $f \in F$ and $z \in \{0,1\}$. A *rule* r is a pair (t, c) where t is a term and $c \in \{0,1\}$. We say that a rule (t, c) is a *c-rule*. We say that a term t (or rule (t, c)) *applies to (or agrees with)* an example e if $e(f) = z$ for every element $(f = z)$ of t. Note that the empty rule applies to any example.

A *decision set* (DS) S is a pair (T, b), where T is a set of terms and $b \in \{0,1\}$ is the classification of the default rule (or the default classification). We denote by $\|S\|$ the size of S which is equal to $(\sum_{t \in T} |t|) + 1$; the $+1$ is for the default rule. The classification function $S : E(F) \to \{0,1\}$ of a DS $S = (T, b)$ is defined by setting $S(e) = b$ for every example $e \in E(F)$ such that no term in T applies to e and otherwise we set $S(e) = 1 - b$.

Decision Lists. A *decision list* (DL) L is a non-empty sequence of rules $(r_1 = (t_1, c_1), \ldots, r_\ell = (t_\ell, c_\ell))$, for some $\ell \geq 0$. The size of a DL L, denoted by $\|L\|$, is equal to $\sum_{i=1}^{\ell}(|t_i| + 1)$. The classification function $L : E(F) \to \{0,1\}$ of a DL L is defined by setting $L(e) = b$ if the first rule in L that applies to e is a b-rule. To ensure that every example obtains some classification, we assume that the term of the last rule is empty and therefore applies to all examples.

r_1 : IF	$(x = 1 \wedge y = 1)$	THEN 0
r_2 : ELSE IF	$(x = 0 \wedge z = 0)$	THEN 1
r_3 : ELSE IF	$(y = 0 \wedge z = 1)$	THEN 0
r_4 : ELSE		THEN 1

Figure 1: Let L be the DL given in the figure and let e be the example given by $e(x) = 0$, $e(y) = 0$ and $e(z) = 1$. Note that $L(e) = 0$. It is easy to verify that $\{y, z\}$ is the only local abductive explanation for e in L of size at most 2. Moreover, both $\{y\}$ and $\{z\}$ are minimal local contrastive explanations for e in L. Let $\tau_1 = \{x \mapsto 1, y \mapsto 1\}$ and $\tau_2 = \{x \mapsto 0, z \mapsto 0\}$ be a partial assignments. Note that τ_1 and τ_2 are minimal global abductive and global contrastive explanations for class 0 w.r.t. L, respectively.

Binary Decision Diagrams. A *binary decision diagram* (BDD) B is a pair (D, ρ) where D is a directed acyclic graph with three special vertices s, t_0, t_1 such that:

- s is a source vertex that can (but does not have to) be equal to t_0 or t_1,
- t_0 and t_1 are the only sink vertices of D,
- every non-sink vertex has exactly two outgoing neighbors, which we call the 0-neighbor and the 1-neighbor, and
- $\rho : V(D) \setminus \{t_0, t_1\} \to F$ is a function that associates with every non-sink node of D a feature in F.

For an example $e \in E$, we denote by $P_B(e)$ (or $P(e)$ if B is clear from the context), the unique path from s to either t_0 or t_1 followed by e in B. That is starting at s and ending at either t_0 or t_1, $P(e)$ is iteratively defined as follows. Initially, we set $P(e) = (s)$, moreover, if $P(e)$ ends in a vertex v other than t_0 or t_1, then we extend $P(e)$ by the $e(\rho(v))$-neighbor of v in D. Let B be a BDD and $e \in E(F)$ be an example. The classification function $B : E(F) \to \{0,1\}$ of B is given by setting $B(e) = b$ if $P_B(e)$ ends in t_b. We denote by $\|B\|$ the size of B, which is equal to $|V(D)|$. We say that B is an OBDD if every path in B contains features in the same order. Moreover, B is a *complete* OBDD if every maximal path contains the same set of features. It is known that every OBDD can be transformed in polynomial-time into an equivalent complete OBDD (Mengel and Slivovsky 2021, Observation 1). All OBDDs considered in the paper are complete.

Ensembles. An *$\mathcal{M}$-ensemble*, also denoted by $\mathcal{M}_{\text{MAJ}}$, $\mathcal{E}$ is a set of models of type $\mathcal{M}$, where $\mathcal{M} \in \{\text{DT}, \text{DS}, \text{DL}, \text{OBDD}\}$. We say that $\mathcal{E}$ classifies an example $e \in E(F)$ as b if so do the majority of models in $\mathcal{E}$, i.e., if there are at least $\lfloor |\mathcal{E}|/2 \rfloor + 1$ models in $\mathcal{E}$ that classify e as b. We denote by $\|\mathcal{E}\|$ the size of $\mathcal{E}$, which is equal to $\sum_{M \in \mathcal{E}} \|M\|$. We additionally consider an *ordered* OBDD-*ensemble*, denoted by $\text{OBDD}^{<}_{\text{MAJ}}$, where all OBDDs in the ensemble respect the same ordering of the features.

3 Considered Problems and Parameters

We consider the following types of explanations (see Marques-Silva's survey (2023)). Let M be a model, e an example over $F(M)$, and let $c \in \{0,1\}$ be a classification (class). We consider the following types of explanations for which an example is illustrated in Figure 1.

- A *(local) abductive explanation (*LAXP*) for e w.r.t. M* is a subset $A \subseteq F(M)$ of features such that $M(e) = M(e')$ for every example e' that agrees with e on A.
- A *(local) contrastive explanation (*LCXP*) for e w.r.t. M* is a set A of features such that there is an example e' such that $M(e') \neq M(e)$ and e' differ from e only on the features in A.
- A *global abductive explanation (*GAXP*) for c w.r.t. M* is a partial example $\tau : F \to \{0,1\}$, where $F \subseteq F(M)$, such that $M(e) = c$ for every example e that agrees with τ.
- A *global contrastive explanation (*GCXP*) for c w.r.t. M* is a partial example $\tau : F \to \{0,1\}$, where $F \subseteq F(M)$, such that $M(e) \neq c$ for every example that agrees with τ.

For each of the above explanation types, each of the considered model types $\mathcal{M}$, and depending on whether or not one wants to find a subset minimal or cardinality-wise minimum explanation, one can now define the corresponding computational problem. For instance:

$\mathcal{M}$-SUBSET-MINIMAL LOCAL ABDUCTIVE EXPLANATION (LAXP$_{\subseteq}$)
INSTANCE: A model $M \in \mathcal{M}$ and an example e.
QUESTION: Find a subset minimal local abductive explanation for e w.r.t. M.

$\mathcal{M}$-CARDINALITY-MINIMAL LOCAL ABDUCTIVE EXPLANATION (LAXP$_{||}$)
INSTANCE: A model $M \in \mathcal{M}$, an example e, and an integer k.
QUESTION: Is there a local explanation for e w.r.t. M of size at most k?

The problems $\mathcal{M}$-$X_{\subseteq}$ and $\mathcal{M}$-$X_{||}$ for $X \in \{$GAXP, LCXP, GCXP$\}$ are defined analogously.

Finally, for these problems, we will consider natural parameters listed in Table 1; not all parameters apply to all considered problems. We denote a problem X parameterized by parameters p, q, r by $X(p+q+r)$.

4 Overview of Results

As we consider several problems, each with several variants and parameters, there are hundreds of combinations to consider. We therefore provide a condensed summary of our results in Tables 2, 3, 4.

The first column in each table indicates whether a result applies to the cardinality-minimal or subset-minimal variant of the explanation problem (i.e., to $X_{\subseteq}$ or $X_{||}$, respectively). The next 4 columns in Tables 2, 3, 4 indicate the parameterization, the parameters are explained in Table 1. A "p" indicates that this parameter is part of the parameterization, a "$-$" indicates that it isn't. A "c" means the parameter is set to a constant, "1" means the constant is 1.

By default, each row in the tables applies to all four problems LAXP, GAXP, GCXP, and LCXP. However, if a result only applies to LCXP, it is stated in parenthesis. So, for instance, the first row of Table 2 indicates that DT-LAXP$_{\subseteq}$, DT-GAXP$_{\subseteq}$, DT-GCXP$_{\subseteq}$, and DT-LCXP$_{\subseteq}$, where the ensemble consists of a single DT, can be solved in polynomial time.

The penultimate row of Table 2 indicates that DT$_{\text{MAJ}}$-LAXP$_{||}$, DT$_{\text{MAJ}}$-GAXP$_{||}$ and DT$_{\text{MAJ}}$-GCXP$_{||}$ are co-NP-hard even if *mnl_size* + *size_elem* + *xp_size* is constant, and DT$_{\text{MAJ}}$-LCXP$_{||}$ is W[1]-hard parameterized by *xp_size* even if *mnl_size* + *size_elem* is constant. Finally, the $\star$ indicates a minor distinction in the complexity between DT-LAXP$_{||}$ and the two problems DT-GAXP$_{||}$ and DT-GCXP$_{||}$. That is, if the cell contains NP-h* or pNP-h*, then DT-LAXP$_{||}$ is NP-hard or pNP-hard, respectively, and neither DT-GAXP$_{||}$ nor DT-GCXP$_{||}$ are in P unless FPT = W[1].

We only state in the tables those results that are not implied by others. Tractability results propagate in the following list from left to right, and hardness results propagate from right to left.

$||$-minimality	$\Rightarrow$	$\subseteq$-minimality
set A of parameters	$\Rightarrow$	set $B \supseteq A$ of parameters
ensemble of models	$\Rightarrow$	single model
unordered OBDD ensemble	$\Rightarrow$	ordered OBDD ensemble

For instance, the tractability of $X_{||}$ implies the tractability of $X_{\subseteq}$, and the hardness of $X_{\subseteq}$ implies the hardness of $X_{||}$.[1]

parameter	definition
ens_size	number of elements of the ensemble
mnl_size	largest number of MNL over all ensemble elem.
terms_elem	largest number of terms per ensemble elem.
term_size	size of a largest term over all ensemble elem.
width_elem	largest width over all ensemble elements
size_elem	size of largest ensemble element
xp_size	size of the explanation

Table 1: Main parameters considered. Note that some parameters (such as *width_elem*) only apply to specific model types.

5 Algorithmic Results

In this section, we will present our algorithmic results. We start with some general observations that are independent of a particular model type.

Theorem 2. *Let $\mathcal{M}$ be any model type such that $M(e)$ can be computed in polynomial-time for $M \in \mathcal{M}$. $\mathcal{M}$-LCXP$_{||}$ parameterized by xp_size is in* XP.

[1] Note that even though the inclusion-wise minimal versions of our problems are defined in terms of finding (instead of decision), the implication still holds because there is a polynomial-time reduction from the finding version to the decision version of LAXP$_{||}$, LCXP$_{||}$, GAXP$_{||}$, and GCXP$_{||}$.

minimality	ens_size	mnl_size	size_elem	xp_size	complexity	result
$\subseteq$	1	–	–	–	P	Thm 10
$=$	1	–	–	–	NP-h*(P)	Thms 8, 32, 33
$=$	1	–	–	p	W[1]-h(P)	Thms 8, 32, 33
$=$	1	–	–	p	XP (P)	Thms 8, 11
$\subseteq$	p	–	–	–	co-W[1]-h(W[1]-h)	Thm 35
$\subseteq$	p	–	–	–	XP	Thm 12
$=$	p	–	–	–	pNP-h*(XP)	Thms 12, 32, 33
$=$	p	p	–	–	FPT	Thm 7
$=$	p	–	p	–	FPT	Thm 7
$=$	p	–	–	c(p)	co-W[1]-h(W[1]-h)	Thm 35
$\subseteq$	–	c	c	–	co-NP-h(NP-h)	Thm 36
$=$	–	c	c	c(p)	co-NP-h(W[1]-h)	Thm 36
$=$	–	–	–	p	co-pNP-h(XP)	Thms 2, 36

Table 2: Explanation complexity when the model is a DT or an ensemble of DTs. See Section 4 for how to read the table.

minimality	ens_size	terms_elem	term_size	xp_size	complexity	result
$\subseteq$	1	–	c	–	co-NP-h(NP-h)	Thm 37
$=$	1	–	–	p	co-pNP-h(W[1]-h)	Thms 37, 38
$=$	c	–	p	p	co-pNP-h(FPT)	Thms 18, 37
$=$	p	p	–	–	FPT	Cor 15
$=$	p	–	c	p	co-pNP-h(W[1]-h)	Thms 37, 39
$\subseteq$	–	c	c	–	co-NP-h(NP-h)	Thm 40
$=$	–	c	c	c(p)	co-NP-h(W[1]-h)	Thm 40
$=$	–	–	–	p	co-pNP-h(XP)	Thms 2, 37

Table 3: Explanation complexity when the model is a DS, a DL, or an ensemble thereof. See Section 4 for how to read the table.

Proof. Let (M, e, k) be the given instance of $\mathcal{M}$-LCXP$_{||}$ and suppose that $A \subseteq F(M)$ is a cardinality-wise minimal local contrastive explanation for e w.r.t. M. Because A is cardinality-wise minimal, it holds the example e_A obtained from e by setting $e_A(f) = 1 - e(f)$ for every $f \in A$ and $e_A(f) = e(f)$ otherwise, is classified differently from e, i.e., $M(e) \neq M(e_A)$. Therefore, a set $A \subseteq F(M)$ is a cardinality-wise minimal local contrastive explanation for e w.r.t. M if and only if $M(e) \neq M(e_A)$ and there is no cardinality-wise smaller set A' for which this is the case. This now allows us to obtain an XP algorithm for $\mathcal{M}$-LCXP$_{||}$ as follows. We first enumerate all possible subsets $A \subseteq F(M)$ of size at most k in time $\mathcal{O}(|F(M)|^k)$ and for each such subset A we test in polynomial-time if $M(e_A) \neq M(e)$. If so, we output that (M, e, k) is a yes-instance and if this is not the case for any of the enumerated subsets, we output correctly that (M, e, k) is a no-instance. □

The remainder of the section is organized as follows. First in Section 5.1, we provide a very general result about Boolean circuits, which will allow us to show a variety of

minimality	ordered/unordered	ens_size	width_elem	size_elem	xp_size	complexity	result
$\subseteq$	u	1	–	–	–	P	Thms 23, 25
$=$	o	1	–	–	–	NP-h(P)	Thms 23, 43
$=$	o	1	–	–	p	W[2]-h(P)	Thms 23, 43
$\subseteq$	u	c	c	–	–	co-NP-h(NP-h)	Thm 41
$=$	u	c	c	–	c(p)	co-NP-h(W[1]-h)	Thm 41
$\subseteq$	o	p	–	–	–	co-W[1]-h(W[1]-h)	Thm 45
$\subseteq$	o	p	–	–	–	XP	Thm 27
$=$	o	p	–	–	–	pNP-h(XP)	Thms 27, 43
$=$	o	p	p	–	–	FPT	Cor 21
$=$	u	p	–	p	–	FPT	Cor 22
$=$	o	p	–	–	c(p)	co-W[1]-h(W[1]-h)	Thm 45
$\subseteq$	o	–	c	c	–	co-NP-h(NP-h)	Thm 44
$=$	o	–	c	c	c(p)	co-NP-h(W[1]-h)	Thm 44
$=$	u	–	–	–	p	co-pNP-h(XP)	Thms 2, 41

Table 4: Explanation complexity when the model is an OBDD or an ensemble thereof. For an ensemble, column "ordered/unordered" indicates whether all the OBDDs in the ensemble have the same variable-order. See Section 4 for how to read the table.

algorithmic results for our models. We then provide our algorithms for the considered models in Subsections 5.2 to 5.4

5.1 A Meta-Theorem for Boolean Circuits

Here, we present our algorithmic result for Boolean circuits that are allowed to employ majority circuits. In particular, we will show that all considered explanation problems are fixed-parameter tractable parameterized by the so-called rankwidth of the Boolean circuit as long as the Boolean circuit uses only a constant number of majority gates. Since our considered models can be naturally translated into Boolean circuits, which require majority gates in the case of ensembles, we will obtain a rather large number of algorithmic consequences from this result by providing suitable reductions of our models to Boolean circuits in the following subsections.

We start by introducing Boolean circuits. A *Boolean circuit (*BC*)* is a directed acyclic graph D with a unique sink vertex o (output gate) such that every vertex $v \in V(D) \setminus \{o\}$ is either:

- an *input gate (IN-gate)* with no incoming arcs,
- an *AND-gate* with at least one incoming arc,
- an *OR-gate* with at least one incoming arcs,
- a *majority-gate (MAJ-gate)* with at least one incoming arc and an integer threshold t_v, or
- a *NOT-gate* with exactly one incoming arc.

We denote by $\mathsf{IG}(D)$ the set of all input gates of D and by MAJ(D) the set of all MAJ-gates of D. For an assignment $\alpha : \mathsf{IG}(D) \to \{0, 1\}$ and a vertex $v \in V(D)$, we denote by $\mathsf{val}(v, D, \alpha)$ the value of the gate v after assigning all input gates according to α. That is, $\mathsf{val}(v, D, \alpha)$ is

recursively defined as follows: If v is an input gate, then $\mathsf{val}(v, D, \alpha) = \alpha(v)$, if v is an AND-gate (OR-gate), then $\mathsf{val}(v, D, \alpha) = \bigwedge_{n \in N_D^-(v)} \mathsf{val}(n, D, \alpha)$ ($\mathsf{val}(v, D, \alpha) = \bigvee_{n \in N_D^-(v)} \mathsf{val}(n, D, \alpha)$), and if v is a MAJ-gate, then $\mathsf{val}(v, D, \alpha) = |\{\, n \mid n \in N_D^-(v) \wedge \mathsf{val}(n, D, \alpha) = 1 \,\}| \geq t_v$. Here and in the following $N_D^-(v)$ denotes the set of all incoming neighbors of v in D. We set $\mathsf{O}(D, \alpha) = \mathsf{val}(o, D, \alpha)$. We say that D is a c-BC if c is an integer and D contains at most c MAJ-gates.

We consider *Monadic Second Order* (MSO_1) logic on structures representing BCs as a directed (acyclic) graph with unary relations to represent the types of gates. That is the structure associated with a given BC D has $V(D)$ as its universe and contains the following unary and binary relations over $V(D)$:

- the unary relations I, A, O, M, and N containing all input gates, all AND-gates, all OR-gates, all MAJ-gates, and all NOT-gates of D, respectively,
- the binary relation Exy containing all pairs $x, y \in V(D)$ such that $(x, y) \in A(D)$.

We assume an infinite supply of *individual variables* and *set variables*, which we denote by lower case and upper case letters, respectively. The available *atomic formulas* are Pg ("the value assigned to variable g is contained in the unary relation or set variable P"), Exy ("vertex x is the head of an edge with tail y"), $x = y$ (equality), and $x \neq y$ (inequality). MSO_1 *formulas* are built up from atomic formulas using the usual Boolean connectives ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$), quantification over individual variables ($\forall x$, $\exists x$), and quantification over set variables ($\forall X$, $\exists X$).

In order to be able to deal with MAJ-gates, we will need a slightly extended version of MSO_1 logic, which we denote by MSOE_1 and which in turn is a slightly restricted version of the so-called distance neighborhood logic that was introduced by Bergougnoux, Dreier, and Jaffke (2023). MSOE_1 extends MSO_1 with *set terms*, which are built from set-variables, unary relations, or other set terms by applying standard set operations such as intersection ($\cap$), union ($\cup$), subtraction ($\setminus$), or complementation (denoted by a bar on top of the term). Note that the distance neighborhood logic introduces neighborhood terms, which extend set terms by allowing an additional neighborhood operator on sets that is not required for our purposes. Like distance neighborhood logic, MSOE_1 also allows for comparisons between set terms, i.e., we can write $t_1 = t_2$ or $t_1 \subseteq t_2$ to express that the set represented by the set term t_1 is equal or a subset of the set represented by the set term t_2, respectively. Most importantly for modelling MAJ-gates is that MSOE_1 allows for *size measurement of terms*, i.e., for a set term t and an integer m, we can write $|t| \geq m$ to express that the set represented by t contains at least m elements.

Let Φ be an MSOE_1 formula (sentence). For a BC D and possibly an additional unary relation $U \subseteq V(D)$, we write $(D, U) \models \Phi$ if Φ holds true on the structure representing D with additional unary relation U. The following proposition is crucial for our algorithms based on MSOE_1 and essentially provides an efficient algorithm for a simple optimization variant of the model checking problem for MSOE_1.

Proposition 3 ((Bergougnoux, Dreier, and Jaffke 2023, Theorem 1.2))**.** *Let D be a c-BC, $U \subseteq V(D)$, and let $\Phi(S_1, \ldots, S_\ell)$ be an MSOE_1 formula with free (non-quantified) set variables $S_1, \ldots, S_\ell$. The problem of computing sets $B_1, \ldots, B_\ell \subseteq V(D)$ such that $(D, U) \models \Phi(B_1, \ldots, B_\ell)$ and $\sum_{i=1}^{\ell} |B_i|$ is minimum is fixed-parameter tractable parameterized by $\mathsf{rw}(D) + |\Phi(S_1, \ldots, S_\ell)|$.*

Theorem 4. *c-BC-LAXP$_{||}$, c-BC-GAXP$_{||}$, c-BC-LCXP$_{||}$, c-BC-GCXP$_{||}$ are fixed-parameter tractable parameterized by the rankwidth of the circuit.*

Proof. Let D be a c-BC with output gate o. We will define one MSOE_1 formula for each of the four considered problems. That is, we will define the $\Phi_{\mathsf{LA}}(S)$, $\Phi_{\mathsf{LC}}(S)$, $\Phi_{\mathsf{GA}}(S_0, S_1)$, and $\Phi_{\mathsf{GC}}(S_0, S_1)$ such that:

- $(D, T) \models \Phi_{\mathsf{LA}}(S)$ if and only if S is a local abductive explanation for e w.r.t. D. Here, e is the given example and T is a unary relation on $V(D)$ given as $T = \{\, v \in \mathsf{IG}(D) \mid e(v) = 1 \,\}$.
- $(D, T) \models \Phi_{\mathsf{LC}}(S)$ if and only if S is a local contrastive explanation for e w.r.t. D. Here, e is the given example and T is a unary relation on $V(D)$ given as $T = \{\, v \in \mathsf{IG}(D) \mid e(v) = 1 \,\}$.
- $(D, C) \models \Phi_{\mathsf{GA}}(S_0, S_1)$ if and only if the partial assignment $\tau : S_0 \cup S_1 \to \{0, 1\}$ with $\tau(s) = 0$ if $s \in S_0$ and $\tau(s) = 1$ if $s \in S_1$ is a global abductive explanation for c w.r.t. D. Here, $c \in \{0, 1\}$ is the given class and C is a unary relation on $V(D)$ that is empty if $c = 0$ and otherwise contains only the output gate o.
- $(D, C) \models \Phi_{\mathsf{GC}}(S_0, S_1)$ if and only if the partial assignment $\tau : S_0 \cup S_1 \to \{0, 1\}$ with $\tau(s) = 0$ if $s \in S_0$ and $\tau(s) = 1$ if $s \in S_1$ is a global contrastive explanation for c w.r.t. D. Here, $c \in \{0, 1\}$ is the given class and C is a unary relation on $V(D)$ that is empty if $c = 0$ and otherwise contains only the output gate o.

Because each of the formulas will have constant length, the theorem then follows immediately from Proposition 3.

We start by defining the auxiliary formula $\mathsf{CON}(A)$ such that $D \models \mathsf{CON}(B)$ if and only if $B = \{\, v \mid v \in V(D) \wedge \mathsf{val}(v, D, \alpha_B) = 1 \,\}$, where $\alpha_B : \mathsf{IG}(D) \to \{0, 1\}$ is the assignment of the input gates of D defined by setting $\alpha_B(v) = 1$ if $v \in B$ and $\alpha_B(v) = 0$, otherwise. In other words $D \models \mathsf{CON}(B)$ if and only if B represents a consistent assignment of the gates, i.e., exactly those gates are in B that are set to 1 if the circuit is evaluated for the input assignment α_B. The formula $\mathsf{CON}(A)$ is equal to $\mathsf{CON}_{\mathrm{MAJ}}(A) \wedge \mathsf{CON}'(A)$, where:

$$\mathsf{CON}_{\mathrm{MAJ}}(A) = \begin{array}{l} \bigwedge_{g' \in \mathrm{MAJ}(D)} g' \in A \leftrightarrow (\exists N\, |N| \geq t_{g'} \wedge \\ (\forall n\, n \in N \leftrightarrow n \in A \wedge \mathrm{IN}(g, n))) \end{array}$$

and

$$\mathsf{CON}'(A) = \begin{array}{ll} \forall g & (\text{AND}(g) \rightarrow \\ & \quad (g \in A \leftrightarrow \forall n\, \text{IN}(g,n) \rightarrow n \in A)) \wedge \\ & (\text{OR}(g) \rightarrow \\ & \quad (g \in A \leftrightarrow \exists n\, \text{IN}(g,n) \wedge n \in A)) \wedge \\ & (\text{NOT}(g) \rightarrow \\ & \quad (g \in A \leftrightarrow \exists n\, \text{IN}(g,n) \wedge n \notin A)) \end{array}$$

We also need the formula $\mathsf{SAT}(E)$ such that $D \models \mathsf{SAT}(B)$ if and only if the assignment $\alpha_B : \mathsf{IG}(D) \rightarrow \{0,1\}$ defined by setting $\alpha_B(v) = 1$ if $v \in B$ and $\alpha_B(v) = 0$ otherwise satisfies the circuit D.

$$\mathsf{SAT}(E) = \exists E'\ E' \cap \mathsf{IG}(D) = E \wedge \mathsf{CON}(E') \wedge o \in E'$$

Finally, we need the formula $\mathsf{AGREE}(E, E_0, E_1)$ such that $D \models \mathsf{AGREE}(B, B_0, B_1)$ if and only if $B_1 \subseteq B$ and $B_0 \cap B = \emptyset$. In other words, the assignment $\alpha_B : \mathsf{IG}(D) \rightarrow \{0,1\}$ defined by setting $\alpha_B(v) = 1$ if $v \in B$ and $\alpha_B(v) = 0$ otherwise is 1 for every feature in B_1 and 0 for every feature in B_0.

$$\mathsf{AGREE}(E, E_0, E_1) = E_1 \subseteq E \wedge E_0 \cap E = \emptyset$$

We are now ready to define the formula $\Phi_{\mathsf{LA}}(S)$.

$$\begin{array}{ll} \Phi_{\mathsf{LA}}(S) = & \exists E_0 \exists E_1\ S = E_0 \cup E_1 \wedge \\ & E_1 \subseteq T \wedge E_0 \subseteq (\mathsf{IG}(D) \setminus T) \wedge \\ & \forall A \subseteq \mathsf{G}(D)\ \mathsf{CON}(A) \rightarrow \\ & (\mathsf{AGREE}(A, E_0, E_1) \rightarrow (\mathsf{SAT}(T) \leftrightarrow o \in A)) \end{array}$$

Note that the set $S = E_0 \cup E_1$ represents the local abductive explanation and, in particular, E_b contains all input gates that are set to b in the explanation. Moreover, the set A represents an example together with its evaluation in the circuit D and the subformula $(\mathsf{SAT}(T) \leftrightarrow o \in A)$ is true if and only if the example represented by A is classified in the same manner as the example represented by T.

We are now ready to define the formula $\Phi_{\mathsf{LC}}(S)$.

$$\begin{array}{ll} \Phi_{\mathsf{LC}}(S) = & S \subseteq \mathsf{IG}(D) \wedge \\ & \exists A' \subseteq \mathsf{G}(D)\ T \setminus S \subseteq A' \wedge S \setminus T \subseteq A' \wedge \\ & A' \subseteq (T \cup S) \setminus (T \cap S) \wedge \\ & \mathsf{CON}(A') \wedge (\mathsf{SAT}(A') \leftrightarrow \neg \mathsf{SAT}(T)) \end{array}$$

Here, S represents the set of features in the local contrastive explanation and A' represents the example e' (together with its evaluation in the circuit D) that differs from the example e (represented by T) only in the features in S and is classified differently from e.

We are now ready to define the formula $\Phi_{\mathsf{GA}}(S_0, S_1)$.

$$\begin{array}{ll} \Phi_{\mathsf{GA}}(S_0, S_1) = & S_0 \subseteq \mathsf{IG}(D) \wedge \\ & S_1 \subseteq \mathsf{IG}(D) \wedge S_0 \cap S_1 = \emptyset \wedge \\ & \forall A \subseteq \mathsf{G}(D)\ \mathsf{CON}(A) \rightarrow \\ & (\mathsf{AGREE}(A, S_0, S_1) \rightarrow (Co \leftrightarrow o \in A)) \end{array}$$

Note that the set $S_0 \cup S_1$ represents the global abductive explanation and, in particular, S_b contains all input gates that are set to b in the explanation. Moreover, the set A represents an example together with its evaluation in the circuit D and the subformula $(Co \leftrightarrow o \in A)$ is true if and only if the example represented by A is classified as c.

We are now ready to define the formula $\Phi_{\mathsf{GC}}(S_0, S_1)$.

$$\begin{array}{ll} \Phi_{\mathsf{GC}}(S_0, S_1) = & S_0 \subseteq \mathsf{IG}(D) \wedge \\ & S_1 \subseteq \mathsf{IG}(D) \wedge S_0 \cap S_1 = \emptyset \wedge \\ & \forall A \subseteq \mathsf{G}(D)\ (\mathsf{CON}(A) \rightarrow \\ & (\mathsf{AGREE}(A, S_0, S_1) \rightarrow (Co \leftrightarrow o \notin A) \end{array}$$

□

5.2 DTs and their Ensembles

Here, we present our algorithms for DTs and their ensembles. We start with a simple translation from DTs to BCs that allow us to employ Theorem 4 for DTs.

Lemma 5. *There is a polynomial-time algorithm that given a* DT $\mathcal{T} = (T, \lambda)$ *and a class* c *produces a circuit* $\mathcal{C}(\mathcal{T}, c)$ *such that:*

(1) for every example e*, it holds that* $\mathcal{T}(e) = c$ *if and only if (the assignment represented by)* e *satisfies* $\mathcal{C}(\mathcal{T}, c)$ *and*

(2) $\mathsf{rw}(\mathcal{C}(\mathcal{T}, c)) \leq 3 \cdot 2^{|\mathsf{MNL}(\mathcal{T})|}$

Proof. Let $\mathcal{T} = (T, \lambda)$ be the given DT and suppose that $\mathsf{MNL}(\mathcal{T})$ is equal to the number of negative leaves; the construction of the circuit $\mathcal{C}(\mathcal{T}, c)$ is analogous if instead $\mathsf{MNL}(\mathcal{T})$ is equal to the number of positive leaves. We first construct the circuit D such that D is satisfied by e if and only if $\mathcal{T}(e) = 0$. D contains one input gate g_f and one NOT-gate $\overline{g_f}$, whose only incoming arc is from g_f, for every feature in $F(\mathcal{T})$. Moreover, for every $l \in L_0(\mathcal{T})$, D contains an AND-gate g_l, whose incoming arcs correspond to the partial assignment $\alpha^l_{\mathcal{T}}$, i.e., for every feature f assigned by $\alpha^l_{\mathcal{T}}$, g_l has an incoming arc from g_f if $\alpha^l_{\mathcal{T}}(f) = 1$ and an incoming arc from $\overline{g_f}$ otherwise. Finally, D contains the OR-gate o, which also serves as the output gate of D, that has one incoming arc from g_l for every $l \in L_0$. This completes the construction of D and it is straightforward to show that D is satisfied by an example e if and only if $\mathcal{T}(e) = 0$. Moreover, using Lemma 1, we obtain that D has treewidth at most $|\mathsf{MNL}(\mathcal{T})|+1$ because the graph obtained from D after removing all gates g_l for every $l \in L_0(\mathcal{T})$ is a tree and therefore has treewidth at most 1. Therefore, using Lemma 1, we obtain that D has rankwidth at most $3 \cdot 2^{|\mathsf{MNL}(\mathcal{T})|}$. Finally, $\mathcal{C}(\mathcal{T}, c)$ can now be obtained from D as follows. If $c = 0$, then $\mathcal{C}(\mathcal{T}, c) = D$. Otherwise, $\mathcal{C}(\mathcal{T}, c)$ is obtained from D after adding one OR-gate that also serves as the new output gate of $\mathcal{C}(\mathcal{T}, c)$ and that has only one incoming arc from o. □

We now provide a translation from DT_{MAJ}s to 1-BCs that will allow us to obtain tractability results for DT_{MAJ}s.

Lemma 6. *There is a polynomial-time algorithm that given a* DT_{MAJ} $\mathcal{F}$ *and a class* c *produces a circuit* $\mathcal{C}(\mathcal{F}, c)$ *such that:*

(1) for every example e*, it holds that* $\mathcal{F}(e) = c$ *if and only if (the assignment represented by)* e *satisfies* $\mathcal{C}(\mathcal{F}, c)$ *and*

(2) $\mathsf{rw}(\mathcal{C}(\mathcal{F}, c)) \leq 3 \cdot 2^{\sum_{\mathcal{T} \in \mathcal{F}} |\mathsf{MNL}(\mathcal{T})|}$

Proof. We obtain the circuit $\mathcal{C}(\mathcal{F}, c)$ from the (not necessarily disjoint) union of the circuits $\mathcal{C}(\mathcal{T}, c)$ for every $\mathcal{T} \in \mathcal{F}$, which we introduced in Lemma 5, after adding a new MAJ-gate with threshold $\lfloor |\mathcal{F}|/2 \rfloor + 1$, which also serves as the output gate of $\mathcal{C}(\mathcal{F}, c)$, that has one incoming arc from the output gate of $\mathcal{C}(\mathcal{T}, c)$ for every $\mathcal{T} \in \mathcal{F}$. Clearly, $\mathcal{C}(\mathcal{F}, c)$ satisfies (1). Moreover, to see that it also satisfies (2), recall that every circuit $\mathcal{C}(\mathcal{T}, c)$ has only $\mathsf{MNL}(\mathcal{T})$ gates apart from the input gates, the NOT-gates connected to the input gates, and the output gate. Therefore, after removing $\mathsf{MNL}(\mathcal{T})$ gates from every circuit $\mathcal{C}(\mathcal{T}, c)$ inside $\mathcal{C}(\mathcal{F}, c)$, the remaining circuit is a tree, which together with Lemma 1 implies (2). □

The following theorem now follows immediately from Theorem 4 together with Lemma 6.

Theorem 7. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. DT_{MAJ}-$\mathcal{P}_{||}$(*ens_size* + *mnl_size*) *and therefore also* DT_{MAJ}-$\mathcal{P}_{||}$(*ens_size* + *size_elem*) *is* FPT.

We now give our polynomial-time algorithms for DTs. We start with the following known result for contrastive explanations.

Theorem 8 ((Barceló et al. 2020, Lemma 14))**.** *There is a polynomial-time algorithm that given a* DT $\mathcal{T}$ *and an example* e *outputs a (cardinality-wise) minimum local contrastive explanation for* e *w.r.t.* $\mathcal{T}$ *or no if such an explanation does not exist. Therefore,* DT-$\text{LCXP}_{||}$ *can be solved in polynomial-time.*

The following auxiliary lemma provides polynomial-time algorithms for testing whether a given subset of features A or partial example e' is a local abductive, global abductive, or global contrastive explanation for a given example e or class c w.r.t. a given DT $\mathcal{T}$.

Lemma 9. *Let* $\mathcal{T}$ *be a* DT*, let* e *be an example and let* c *be a class. There are polynomial-time algorithms for the following problems:*

(1) Decide whether a given subset $A \subseteq F(\mathcal{T})$ *of features is a local abductive explanation for* e *w.r.t.* $\mathcal{T}$*.*

(2) Decide whether a given partial example e' *is a global abductive/contrastive explanation for* c *w.r.t.* $\mathcal{T}$*.*

Proof. Let $\mathcal{T}$ be a DT, let e be an example and let c be a class. Note that we assume here that $\mathcal{T}$ does not have any contradictory path, i.e., a root-to-leaf path that contains that assigns any feature more than once. Because if this was not the case, we could easily simplify $\mathcal{T}$ in polynomial-time.

We start by showing (1). A subset $A \subseteq F(\mathcal{T})$ of features is a local abductive explanation for e w.r.t. $\mathcal{T}$ if and only if the DT $\mathcal{T}_{|e_{|A}}$ does only contain $\mathcal{T}(e)$-leaves, which can clearly be decided in polynomial-time. Here, $e_{|A}$ is the partial example equal to the restriction of e to A. Moreover, $\mathcal{T}_{|e'}$ for a partial example e' is the DT obtained from $\mathcal{T}$ after removing every $1 - e'(f)$-child from every node t of $\mathcal{T}$ assigned to a feature f for which e' is defined.

Similarly, for showing (2), observe that partial example (assignment) $\tau : F \to \{0, 1\}$ is a global abductive explanation for c w.r.t. $\mathcal{T}$ if and only if the DT $\mathcal{T}_{|\tau}$ does only contain c-leaves, which can clearly be decided in polynomial-time.

Finally, note that a partial example $\tau : F \to \{0, 1\}$ is a global contrastive explanation for c w.r.t. $\mathcal{T}$ if and only if the DT $\mathcal{T}_{|\tau}$ does not contain any c-leaf, which can clearly be decided in polynomial-time. □

Using dedicated algorithms for the inclusion-wise minimal variants of LAXP, GAXP, and GCXP together with Theorem 8, we obtain the following result.

Theorem 10. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. DT-$\mathcal{P}_{\subseteq}$ *can be solved in polynomial-time.*

Proof. Note that the statement of the theorem for DT-$\text{LCXP}_{\subseteq}$ follows immediately from Theorem 8. Therefore, it suffices to show the statement of the theorem for the remaining 3 problems.

Let $(\mathcal{T}, e)$ be an instance of DT-$\text{LAXP}_{\subseteq}$. We start by setting $A = F(\mathcal{T})$. Using Lemma 9, we then test for any feature f in A, whether $A \setminus \{f\}$ is still a local abductive explanation for e w.r.t. $\mathcal{T}$ in polynomial-time. If so, we repeat the process after setting A to $A \setminus \{f\}$ and otherwise we do the same test for the next feature $f \in A$. Finally, if $A \setminus \{f\}$ is not a local abductive explanation for every $f \in A$, then A is an inclusion-wise minimal local abductive explanation and we can output A.

The polynomial-time algorithm for an instance $(\mathcal{T}, c)$ of DT-$\text{GAXP}_{\subseteq}$ nowworks as follows. Let l be a c-leaf of $\mathcal{T}$; if no such c-leaf exists, then we can correctly output that there is no global abductive explanation for c w.r.t. $\mathcal{T}$. Then, $\alpha^l_{\mathcal{T}}$ is a global abductive explanation for c w.r.t. $\mathcal{T}$. To obtain an inclusion-wise minimal solution, we do the following. Let $F = F(\alpha^l_{\mathcal{T}})$ be the set of features on which $\alpha^l_{\mathcal{T}}$ is defined. We now test for every feature $f \in F$ whether the restriction $\alpha^l_{\mathcal{T}}[F \setminus \{f\}]$ of $\alpha^l_{\mathcal{T}}$ to $F \setminus \{f\}$ is a global abductive explanation for c w.r.t. $\mathcal{T}$. This can clearly be achieved in polynomial-time with the help of Lemma 9. If this is true for any feature $f \in F$, then we repeat the process for $\alpha^l_{\mathcal{T}}[F \setminus \{f\}]$, otherwise we output $\alpha^l_{\mathcal{T}}$. A very similar algorithm now works for the DT-$\text{GCXP}_{\subseteq}$ problem, i.e., instead of starting with a c-leaf l we start with a c'-leaf, where $c \neq c'$. □

The following theorem uses an exhaustive enumeration of all possible explanations together with Lemma 9 to check whether a set of features or a partial example is an explanation.

Theorem 11. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. DT-$\mathcal{P}_{||}$(*xp_size*) *is in* XP.

Proof. We start by showing the statement of the theorem for DT-$\text{LAXP}_{||}$. Let $(\mathcal{T}, e, k)$ be an instance of DT-$\text{LAXP}_{||}$. We first enumerate all subsets $A \subseteq F(\mathcal{T})$ of size at most k in time $\mathcal{O}(|F(\mathcal{T})|^k)$. For every such subset A, we then test whether A is a local abductive explanation for e w.r.t. $\mathcal{T}$ in polynomial-time with the help of Lemma 9. If so, we output A as the solution. Otherwise, i.e., if no such subset is a local

abductive explanation for e w.r.t. $\mathcal{T}$, we output correctly that $(\mathcal{T}, e, k)$ has no solution.

Let $(\mathcal{T}, c, k)$ be an instance of DT-GAXP$_{||}$. We first enumerate all subsets $A \subseteq F(\mathcal{T})$ of size at most k in time $\mathcal{O}(|F(\mathcal{T})|^k)$. For every such subset A, we then enumerate all of the at most $2^{|A|} \leq 2^k$ partial examples (assignments) $\tau : A \to \{0, 1\}$ in time $\mathcal{O}(2^k)$. For every such partial example τ, we then use Lemma 9 to test whether τ is a global abductive explanation for c w.r.t. $\mathcal{T}$ in polynomial-time. If so, we output e as the solution. Otherwise, i.e., if no such partial example is a global abductive explanation for c w.r.t. $\mathcal{T}$, we output correctly that $(\mathcal{T}, c, k)$ has no solution. The total runtime of the algorithm is at most $2^k|F(\mathcal{T})|^k|\mathcal{T}|^{\mathcal{O}(1)}$.

The algorithm for DT-GCXP$_{||}$ is now very similar to the above algorithm for DT-GAXP$_{||}$. □

The next theorem uses our result that the considered problems are in polynomial-time for DTs together with an XP-algorithm that transforms any DT$_{\text{MAJ}}$ into an equivalent DT.

Theorem 12. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. DT$_{\text{MAJ}}$-$\mathcal{P}_{\subseteq}$(*ens_size*) *and* DT$_{\text{MAJ}}$-LCXP$_{||}$(*ens_size*) *are in* XP.

Proof. Let $\mathcal{F} = \{\mathcal{T}_1, \ldots, \mathcal{T}_\ell\}$ be the random forest given as an input to any of the five problems stated above. We start by constructing a DT $\mathcal{T}$ that is not too large (of size at most $m^{|\mathcal{F}|}$, where $m = (\max\{\, |L(\mathcal{T}_i)| \mid 1 \leq i \leq \ell \,\}))$ and that is equivalent to $\mathcal{F}$ in the sense that $\mathcal{F}(e) = \mathcal{T}(e)$ for every example in time at most $\mathcal{O}(m^{|\mathcal{F}|})$. Since all five problems can be solved in polynomial-time on $\mathcal{T}$ (because of Theorem 10) this completes the proof of the theorem.

We construct $\mathcal{T}$ iteratively as follows. First, we set $\mathcal{T}'_1 = \mathcal{T}_1$. Moreover, for every $i > 1$, $\mathcal{T}'_i$ is obtained from $\mathcal{T}'_{i-1}$ and $\mathcal{T}_i$ by making a copy $\mathcal{T}_i^l$ of $\mathcal{T}_i$ for every leaf l of $\mathcal{T}'_{i-1}$ and by identifying the root of $\mathcal{T}_i^l$ with the leaf l of $\mathcal{T}'_{i-1}$. Then, $\mathcal{T}$ is obtained from $\mathcal{T}'_\ell$ changing the label of every leaf l be a leaf of $\mathcal{T}'_\ell$ as follows. Let P be the path from the root of $\mathcal{T}'_\ell$ to l in $\mathcal{T}'_\ell$. By construction, P goes through one copy of $\mathcal{T}_i$ for every i with $1 \leq i \leq \ell$ and therefore also goes through exactly one leaf of every $\mathcal{T}_i$. We now label l according to the majority of the labels of these leaves. This completes the construction of $\mathcal{T}$ and it is easy to see that $\mathcal{F}(e) = \mathcal{T}(e)$ for every example e. Moreover, the size of $\mathcal{T}$ is at most $\Pi_{i=1}^{\ell}|L(\mathcal{T}_i)| \leq m^\ell$, where $m = (\max\{\, |L(\mathcal{T}_i)| \mid 1 \leq i \leq \ell \,\})$ and $\mathcal{T}$ can be constructed in time $\mathcal{O}(m^\ell)$. □

5.3 DSs, DLs and their Ensembles

This subsection is devoted to our algorithmic results for DS, DLs and their ensembles. Our first algorithmic result is again based on our meta-theorem (Theorem 4) and a suitable translation from DS$_{\text{MAJ}}$ and DL$_{\text{MAJ}}$ to a Boolean circuit.

Lemma 13. *There is a polynomial-time algorithm that given a* DS/DL L *and a class* c *produces a circuit* $\mathcal{C}(L, c)$ *such that:*

- *for every example* e*, it holds that* $L(e) = c$ *if and only if* e *satisfies* $\mathcal{C}(L, c)$
- $\mathsf{rw}(\mathcal{C}(L, c)) \leq 3 \cdot 2^{3|L|}$

Proof. Since every DS can be easily transformed into a DL with the same number of terms, it suffices to show the lemma for DLs. Let L be a DL with rules $(r_1 = (t_1, c_1), \ldots, r_\ell = (t_\ell, c_\ell))$. We construct the circuit $D = \mathcal{C}(L, c)$ as follows. D contains one input gate g_f and one NOT-gate $\overline{g_f}$, whose only incoming arc is from g_f, for every feature in $F(L)$. Furthermore, for every rule $r_i = (t_i, c_i)$, D contains an AND-gate g_{r_i}, whose in-neighbors are the literals in t_i, i.e., if t_i contains a literal $f = 0$, then g_{r_i} has $\overline{g_f}$ as an in-neighbor and if t_i contains a literal $f = 1$, then g_{r_i} has g_f as an in-neighbor. We now split the sequence $\rho = (r_1 = (t_1, c_1), \ldots, r_\ell = (t_\ell, c_\ell))$ into (inclusion-wise) maximal consecutive subsequences ρ_i of rules that have the same class. Let $(\rho_1, \ldots, \rho_r)$ be the sequence of subsequences obtained in this manner, i.e., $\rho_1 \circ \rho_2 \circ \cdots \circ \rho_r = \rho$, every subsequence ρ_i is non-empty and contains only rules from one class, and the class of any rule in ρ_i is different from the class of any rule in ρ_{i+1} for every i with $1 \leq i < r$. Now, for every subsequence ρ_i, we add an OR-gate g_{ρ_i} to D, whose in-neighbors are the gates g_r for every rule r in ρ_i. Let C be the set of all subsequences ρ_i that only contains rules with class c and let $\overline{C}$ be the set of all other subsequences, i.e., those that contain only rules whose class is not equal to c. For every subsequence ρ in $\overline{C}$, we add a NOT-gate $\overline{g_\rho}$ to D, whose in-neighbor is the gate g_ρ. Moreover, for every subsequence ρ_i in C, we add an AND-gate $g^A_{\rho_i}$ to D whose in-neighbors are g_{ρ_i} as well as $\overline{g_{\rho_j}}$ for every $\rho_j \in \overline{C}$ with $j < i$. Finally, we add an OR-gate o to D that also serves as the output gate of D and whose in-neighbors are all the gates g^A_ρ for every $\rho \in C$. This completes the construction of D and it is easy to see that $L(e) = c$ if and only if e satisfies D for every example e, which shows that D satisfies (1). Towards showing (2), let G be the set of all gates in D apart from the gates o and g_f and $\overline{g_f}$ for every feature f. Then, $|G| \leq 3|L|$ and $D \setminus G$ is a forest, which together with Lemma 1 implies (2). □

Lemma 14. *Let* $\mathcal{M} \in \{\text{DS}_{\text{MAJ}}, \text{DL}_{\text{MAJ}}\}$. *There is a polynomial-time algorithm that given an* $\mathcal{M}$ $\mathcal{L}$ *and a class* c *produces a circuit* $\mathcal{C}(\mathcal{L}, c)$ *such that:*

(1) for every example e*, it holds that* $\mathcal{L}(e) = c$ *if and only if* e *satisfies* $\mathcal{C}(\mathcal{L}, c)$

(2) $\mathsf{rw}(\mathcal{C}(\mathcal{L}, c)) \leq 3 \cdot 2^{3\sum_{L \in \mathcal{L}} |L|}$

Proof. We obtain the circuit $\mathcal{C}(\mathcal{L}, c)$ from the (not necessarily disjoint) union of the circuits $\mathcal{C}(L, c)$, which are provided in Lemma 13, for every $L \in \mathcal{L}$ after adding a new MAJ-gate with threshold $\lfloor |\mathcal{L}|/2 \rfloor + 1$, which also serves as the output gate of $\mathcal{C}(\mathcal{L}, c)$, that has one incoming arc from the output gate of $\mathcal{C}(L, c)$ for every $L \in \mathcal{L}$. Clearly, $\mathcal{C}(\mathcal{L}, c)$ satisfies (1). Moreover, to see that it also satisfies (2), recall that every circuit $\mathcal{C}(L, c)$ has only $3|L|$ gates apart from the input gates, the NOT-gates connected to the input gates, and the output gate. Therefore, after removing $3|L|$ gates from every circuit $\mathcal{C}(L, c)$ inside $\mathcal{C}(\mathcal{L}, c)$, the remaining circuit is a tree, which together with Lemma 1 implies (2). □

The following corollary now follows immediately from Lemma 14 and Theorem 4.

Corollary 15. *Let* $\mathcal{M} \in \{\mathrm{DS}_{\mathrm{MAJ}}, \mathrm{DL}_{\mathrm{MAJ}}\}$ *and let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. $\mathcal{M}$-$\mathcal{P}_{||}$(*ens_size* + *terms_elem*) *is* FPT.

Unlike, DTs, where DT-LCXP$_{||}$ is solvable in polynomial-time, this is not the case for DS-LCXP$_{||}$. Nevertheless, we are able to provide the following result, which shows that DS-LCXP$_{||}$ (and even DL-LCXP$_{||}$) is fixed-parameter tractable parameterized by *term_size* and *xp_size*. The algorithm is based on a novel characterization of local contrastive explanations for DLs.

Lemma 16. *Let* $\mathcal{M} \in \{\mathrm{DS}, \mathrm{DL}\}$. $\mathcal{M}$-LCXP$_{||}$ *for* $M \in \mathcal{M}$ *and integer* k *can be solved in time* $\mathcal{O}(a^k \|M\|^2)$*, where* a *is equal to term_size.*

Proof. Since any DS can be easily translated into a DL without increasing the size of any term, it suffices to show the lemma for DLs. Let (L, e, k) be an instance of DL-LCXP$_{||}$, where $L = (r_1 = (t_1, c_1), \ldots, r_\ell = (t_\ell, c_\ell))$ is a DL, and let r_i be the rule that classifies e, i.e., the first rule that applies to e.

Let R be the set of all rules r_j of L with $c_j \neq c_i$. For a rule $r \in R$, let $A \subseteq F(L)$ such that the example e_A, i.e., the example obtained from e after setting $e_A(f) = 1 - e(f)$ for every $f \in A$ and $e_A(f) = e(f)$ otherwise, is classified by rule r. We claim that:

(1) For every $r \in R$ and every set $A \subseteq F(\mathcal{T})$ such that e_A is classified by r, it holds that A is a local contrastive explanation for e w.r.t. L.

(2) Every local contrastive explanation A for e w.r.t. L contains a subset $A' \subseteq A$ for which there is a rule $r \in R$ such that $e_{A'}$ is classified by r.

Towards showing (1), let $r \in R$ and let $A \subseteq F(\mathcal{T})$ such that e_A is classified by r. Then, e_A differs from e only on the features in A and moreover $M(e) \neq M(e_A)$ because $r \in R$. Therefore, A is a local contrastive explanation for e w.r.t. $\mathcal{T}$.

Towards showing (2), let $A \subseteq F(A)$ be a local contrastive explanation for e w.r.t. $\mathcal{T}$. Then, there is an example e' that differs from e only on some set $A' \subseteq A$ of features such that e' is classified by a rule $r \in R$. Then, $e_{A'}$ is classified by r, showing (2).

Note that due to (1) and (2), we can compute a smallest local contrastive explanation for e w.r.t. L by the algorithm illustrated in Algorithm 1. That is, the algorithm computes the smallest set A of at most k features such that e_A is classified by r_j for every rule $r_j \in R$. It then, returns the smallest such set over all rules r_j in R if such a set existed for at least one of the rules in R. The main ingredient of the algorithm is the function FINDLCEXFORRULE(L, e, r_j), which is illustrated in Algorithm 2, and that for a rule $r_j \in R$ computes the smallest set A of at most k features such that e_A is classified by r_j. The function Algorithm 2 achieves this as follows. It first computes the set $A_0 = \{ f \mid (f = 1 - e(f)) \in t_j \}$ of features that need to be part of A in order for e_A to satisfy r_j, i.e., all features f where $e(f)$ differs from the literal in t_j. It then computes

Algorithm 1 Algorithm used in Lemma 16 to compute a local contrastive explanation for example e w.r.t. a DL L of size at most k if such an explanation exists. The algorithm uses the function **findLCEXForRule**(L, e, r_j) as a subroutine, which is illustrated in Algorithm 2

Input: DL $L = (r_1 = (t_1, c_1), \ldots, r_\ell = (t_\ell, c_\ell))$, example e and integer k (global variable).
Output: return a cardinality-wise minimum local contrastive explanation $A \subseteq F(L)$ with $|A| \leq k$ for e w.r.t. L or `nil` if such an explanation does not exists.

1: **function** FINDLCEX(L, e)
2: $r_i \leftarrow$ rule of L that classifies e
3: $R \leftarrow \{ r_j \in L \mid c_j \neq c_i \}$
4: $A_b \leftarrow$ `nil`
5: **for** $r_j \in R$ **do**
6: $A \leftarrow$ FINDLCEXFORRULE(L, e, r_j)
7: **if** $A \neq$ `nil` and ($A_b =$ `nil` or $|A_b| > |A|$) **then**
8: $A_b \leftarrow A$
9: **return** A_b

the set A recursively via the (bounded-depth) branching algorithm given in FINDLCEXFORRULEREC(L, e, r_j, A), which given a set A' of features such that $e_{A'}$ satisfies r_j computes a smallest extension (superset) A of A' of size at most k such that e_A is classified by r_j. To do so the function first checks in Line 5 whether $|A'| > k$ and if so correctly returns `nil`. Otherwise, the function checks whether there is any rule r_ℓ with $\ell < j$ that is satisfied by $e_{A'}$. If that is not the case, it correctly returns A' (in Line 9). Otherwise, the function computes the set B of features in Line 11 that occur in t_ℓ – and therefore can be used to falsify r_ℓ – but do not occur in A' or in t_j. Note that we do not want to include any feature in A' or t_j in B since changing those features would either contradict previous branching decisions made by our algorithm or it would prevent us from satisfying r_j. The function then returns `nil` if the set B is empty in Line 13, since in this case it is no longer possible to falsify the rule r_ℓ. Otherwise, the algorithm branches on every feature in $f \in B$ and tries to extend A' with f using a recursive call to itself with A' replaced by $A' \cup \{f\}$ in Line 16. It then returns the best solution found by any of those recursive calls in Line 19. This completes the description of the algorithm, which can be easily seen to be correct using (1) and (2).

We are now ready to analyze the runtime of the algorithm, i.e., Algorithm 1. First note that all operations in Algorithm 1 and Algorithm 2 that are not recursive calls take time at most $\mathcal{O}(\|L\|)$. We start by analyzing the run-time of the function FINDLCEXFORRULE(L, e, r_j), which is at most $\mathcal{O}(\|L\|)$ times the number of recursive calls to the function FINDLCEXFORRULEREC(L, e, r_j, A), which in turn can be easily seen to be at most a^r since the function branches into at most a branches at every call and the depth of the branching is at most k. Therefore, we obtain $\mathcal{O}(a^r \|L\|)$ as the total runtime of the function FINDLCEXFORRULE(L, e, r_j), which implies a total runtime of the algorithm of $\mathcal{O}(a^r \|L\|^2)$. □

The following lemma is now a natural extension of Lemma 16 for ensembles of DLs.

Algorithm 2 Algorithm used as a subroutine in Algorithm 1 to compute a smallest set A of at most k features such that e_A is classified by the rule r_j of a DL L.

Input: DL $L = (r_1 = (t_1, c_1), \ldots, r_\ell = (t_\ell, c_\ell))$, example e, rule r_j, and integer k (global variable).
Output: return a smallest set $A \subseteq F(L)$ with $|A| \leq k$ such that e_A is classified by r_j if such a set exists and otherwise return `nil`.

1: **function** FINDLCEXFORRULE(L, e, r_j)
2: $A_0 \leftarrow \{ f \mid (f = 1 - e(f)) \in t_j \}$
3: **return** FINDLCEXFORRULEREC(L, e, r_j, A_0)
4: **function** FINDLCEXFORRULEREC(L, e, r_j, A')
5: **if** $|A'| > k$ **then**
6: **return** `nil`
7: $r_\ell \leftarrow$ any rule r_ℓ with $\ell < j$ that is satisfied by $e_{A'}$
8: **if** $r_\ell = \texttt{nil}$ **then**
9: **return** A'
10: $F_j \leftarrow \{ f \mid (f = b) \in t_j \wedge b \in \{0, 1\} \}$
11: $B \leftarrow \{ f \mid (f = e_{A'}) \in t_\ell \} \setminus (A' \cup F_j)$
12: **if** $B = \emptyset$ **then**
13: **return** `nil`
14: $A_b \leftarrow \texttt{nil}$
15: **for** $f \in B$ **do**
16: $A \leftarrow$FINDLCEXFORRULEREC(L, e, r_j, $A' \cup \{f\}$)
17: **if** $A \neq \texttt{nil}$ and $|A| \leq k$ **then**
18: **if** ($A_b == \texttt{nil}$ or $|A_b| > |A|$) **then**
19: $A_b \leftarrow A$
return A_b

Lemma 17. *Let $\mathcal{M} \in \{\text{DS}_{\text{MAJ}}, \text{DL}_{\text{MAJ}}\}$. $\mathcal{M}$-LCXP$_{||}$ for $M \in \mathcal{M}$ and integer k can be solved in time $\mathcal{O}(m^s a^k \|M\|^2)$, where m is terms_elem, s is ens_size, and a is term_size.*

Proof. Since any DS_{MAJ} can be easily translated into a DL_{MAJ} without increasing the size of any term and where every ensemble element has at most one extra rule using the empty term, it suffices to show the lemma for DL_{MAJ}s. The main ideas behind the algorithm, which is illustrated in Algorithm 3, are similar to the ideas behind the algorithm used in Lemma 16 for a single DL.

Let $(\mathcal{L}, e, k)$ be an instance of DL_{MAJ}-LCXP$_{||}$ DL_{MAJ} with $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ and $L_i = (r_1^i = (t_1^i, c_1^i), \ldots, r_\ell^i = (t_\ell^i, c_\ell^i))$ for every $i \in [\ell]$. First note that if A is a local contrastive explanation for e w.r.t. $\mathcal{L}$, then there is an example e' that differs from e only on the features in A such that $\mathcal{L}(e) \neq \mathcal{L}(e')$. Clearly e' is classified by exactly one rule of every DL L_i. The first idea behind Algorithm 3 is therefore to enumerate all possibilities for the rules that classify e'; this is done in Line 5 of the algorithm. Clearly, only those combinations of rules are relevant that lead to a different classification of e' compared to e and this is ensured in Line 8 of the algorithm. For every such combination $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$ of rules the algorithm then calls the subroutine FINDELCEXR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$), which is illustrated in Algorithm 4, and computes the smallest set A of at most k features such that e_A is classified by the rule $r_{j_o}^o$ of L_o for every $o \in [\ell]$. This subroutine and the proof of its correctness work very similar to the subroutine **findLCEXForRule**(L, e, r_j) of Algorithm 2 used in Lemma 16 for a single DL and is therefore not repeated here. In essence the subroutine branches over all possibilities for such a set A.

Algorithm 3 Algorithm used in Lemma 17 to compute a local contrastive explanation for example e w.r.t. a DL_{MAJ} $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ with $L_i = (r_1^i = (t_1^i, c_1^i), \ldots, r_\ell^i = (t_\ell^i, c_\ell^i))$ of size at most k if such an explanation exists. The algorithm uses the function **findELCEXForRules**($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$) as a subroutine, which is illustrated in Algorithm 4.

Input: DL_{MAJ} $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ with $L_i = (r_1^i = (t_1^i, c_1^i), \ldots, r_\ell^i = (t_\ell^i, c_\ell^i))$, example e and integer k (global variable).
Output: return a cardinality-wise minimum local contrastive explanation $A \subseteq F(\mathcal{L})$ with $|A| \leq k$ for e w.r.t. $\mathcal{L}$ or `nil` if such an explanation does not exists.

1: **function** FINDELCEX($\mathcal{L}$, e)
2: **for** $o \in [k]$ **do**
3: $R^o \leftarrow \{ r \in L_o \}$
4: $A_b \leftarrow \texttt{nil}$
5: **for** $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell) \in R^1 \times \cdots \times R^\ell$ **do**
6: $n_{\neq} \leftarrow |\{ o \in [\ell] \mid c_{j_o}^o \neq \mathcal{L}(e) \}|$
7: $n_{=} \leftarrow |\{ o \in [\ell] \mid c_{j_o}^o = \mathcal{L}(e) \}|$
8: **if** $n_{\neq} > n_{=}$ **then**
9: $A \leftarrow$ FINDELCEXR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$)
10: **if** $A \neq \texttt{nil}$ and ($A_b = \texttt{nil}$ or $|A_b| > |A|$) **then**
11: $A_b \leftarrow A$
12: **return** A_b

We are now ready to analyze the runtime of the algorithm, i.e., Algorithm 3. First note that the function FINDELCEX($\mathcal{L}$, e) makes at most m^s calls to the subroutine FINDELCEXR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$) and apart from those calls all operations take time at most $\mathcal{O}(\|\mathcal{L}\|)$. Moreover, the runtime of the subroutine FINDELCEXR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$) is the same as the runtime of the subroutine FINDLCEXFORRULE(L, e, r_j) given in Algorithm 2, i.e., $\mathcal{O}(a^k \|\mathcal{L}\|^2)$. Therefore, we obtain $\mathcal{O}(m^s a^k \|\mathcal{L}\|^2)$ as the total run-time of the algorithm. □

The following theorem now follows immediately from Lemma 17.

Theorem 18. *Let $\mathcal{M} \in \{\text{DS}, \text{DL}\}$. $\mathcal{M}$-LCXP$_{||}$(terms_elem + xp_size) is* FPT*, when ens_size is constant.*

5.4 OBDDs and their Ensembles

In this subsection, we will present our algorithmic results for OBDDs and their ensembles $\text{OBDD}_{\text{MAJ}}^{<}$ and OBDD_{MAJ}. Interestingly, while seemingly more powerful OBDDs and $\text{OBDD}_{\text{MAJ}}^{<}$s behave very similar to DTs and DT_{MAJ}s if one replaces *mnl_size* with *width_elem*. On the other hand, allowing different orderings for every ensemble OBDD makes OBDD_{MAJ}s much more powerful and harder to explain (see Section 6.3 for an explanation of this phenomenon).

We start by providing reductions of OBDDs and $\text{OBDD}_{\text{MAJ}}^{<}$s to Boolean circuits, which will allow us to employ Theorem 4. The following lemma follows from (Jha

Algorithm 4 Algorithm used as a subroutine in Algorithm 3 to compute a smallest set A of at most k features such that e_A is classified by the rule r_j^o for every DL L_o.

Input: DL $\mathrm{DL}_{\mathrm{MAJ}}$ $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ with $L_i = (r_1^i = (t_1^i, c_1^i), \ldots, r_\ell^i = (t_\ell^i, c_\ell^i))$, example e, rules $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$, and integer k (global variable).

Output: return a smallest set $A \subseteq F(L)$ with $|A| \leq k$ such that e_A is classified by $r_{j_o}^o$ for every DL L_o if such a set exists and otherwise return `nil`.

function FINDELCEXR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$)
  $A_0 \leftarrow \{ f \mid (f = 1 - e(f)) \in t_{j_o}^o \wedge o \in [\ell] \}$
  return FINDELCEXRR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$, A_0)
function FINDELCEXRR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$, A')
  if $|A'| > k$ **then**
    return `nil`
  $r_\ell^o \leftarrow$ any rule with $\ell < j_o$ that is satisfied by $e_{A'}$ for $o \in [\ell]$
  if $r_\ell^o = \mathtt{nil}$ **then**
    return A'
  $F' \leftarrow \{ f \mid (f = b) \in t_{j_o}^o \wedge o \in [\ell] \wedge b \in \{0, 1\} \}$
  $B \leftarrow \{ f \mid (f = e_{A'}) \in t_\ell^o \} \setminus (A' \cup F')$
  if $B = \emptyset$ **then**
    return `nil`
  $A_b \leftarrow$ `nil`
  for $f \in B$ **do**
    $A \leftarrow$FINDELCEXRR($\mathcal{L}$, e, $(r_{j_1}^1, \ldots, r_{j_\ell}^\ell)$, $A' \cup \{f\}$)
    if $A \neq$ `nil` and $|A| \leq k$ **then**
      if ($A_b ==$ `nil` or $|A_b| > |A|$) **then**
        $A_b \leftarrow A$
  return A_b

and Suciu 2012, Lemma 4.1) since the rank-width is upper bounded by the path-width.

Lemma 19 ((Jha and Suciu 2012, Lemma 4.1))**.** *There is a polynomial-time algorithm that given a* OBDD O *and a class* c *produces a circuit* $\mathcal{C}(O, c)$ *such that:*

- *for every example* e*, it holds that* $O(e) = c$ *if and only if* e *satisfies* $\mathcal{C}(O, c)$
- $\mathsf{rw}(\mathcal{C}(O, c)) \leq 5\mathsf{width}(O)$

Lemma 20. *There is a polynomial-time algorithm that given an* $\mathrm{OBDD}^<_{\mathrm{MAJ}}$ $\mathcal{O}$ *and a class* c *produces a circuit* $\mathcal{C}(\mathcal{O}, c)$ *such that:*

(1) for every example e*, it holds that* $\mathcal{O}(e) = c$ *if and only if* e *satisfies* $\mathcal{C}(\mathcal{O}, c)$

(2) $\mathsf{rw}(\mathcal{C}(\mathcal{O}, c)) \leq 3 \cdot 2^{|\mathcal{O}|5 \max_{O \in \mathcal{O}} \mathsf{width}(O)}$

Proof. We obtain the circuit $\mathcal{C}(\mathcal{O}, c)$ from the (not necessarily disjoint) union of the circuits $\mathcal{C}(O, c)$, which are provided in Lemma 19, for every $O \in \mathcal{O}$ after adding a new MAJ-gate r with threshold $\lfloor |\mathcal{O}|/2 \rfloor + 1$, which also serves as the output gate of $\mathcal{C}(\mathcal{O}, c)$, that has one incoming arc from the output gate of $\mathcal{C}(O, c)$ for every $O \in \mathcal{O}$. Clearly, $\mathcal{C}(\mathcal{O}, c)$ satisfies (1). Moreover, to see that it also satisfies (2), we first need to provide the construction for $\mathcal{C}(O, c)$ given in (Jha and Suciu 2012, Lemma 4.1).

That is, $\mathcal{C}(O, c)$ for a given complete OBDD $O = (D, \rho)$ and a given class c is defined as follows. The main idea behind the construction is to identify every vertex v of D with a propositional formula F_v as follows. First, we set F_{t_b} to be true if and only if $b = c$ for the two sink vertices t_0 and t_1 of D. Moreover, if v is an inner vertex of D with $f = \rho(v)$, 0-neighbor n_0, and 1-neighbor n_1, the formula F_v is defined as $(f = 0 \wedge F_{n_0}) \vee (f = 1 \wedge F_{n_1})$. Then, it holds that $M(e) = c$ if and only if (the assignment corresponding to) e satisfies the formula F_s for the root s of D. The circuit $C = \mathcal{C}(O, c)$ that corresponds to F_s can now be obtained from $D \setminus \{t_0, t_1\}$ by applying the following modifications:

- We add one input gate g_f for every feature $f \in F(\mathcal{O})$.
- We add one NOT-gate $\overline{g_f}$ for every feature $f \in F(O)$, whose only incoming arc is from g_f.
- We replace every inner vertex v of D with an OR-gate o_v.
- We replace every arc $e = (u, v)$ with $f = \rho(u)$ of $D \setminus \{t_0, t_1\}$ with an AND-gate a_e that has one outgoing arc to o_u and that has one incoming arc from o_v and one incoming arc from g_f ($\overline{g_f}$) if v is a 1-neighbor (0-neighbor) of u in D.
- We replace every arc $e = (u, t_c)$ with $f = \rho(u)$ of D with an arc from g_f ($\overline{g_f}$) to o_u if t_c is a 1-neighbor (0-neighbor) of u in D.
- We remove every arc $e = (u, t_{c'})$ with $c' \neq c$ from D.
- We let o_s be the root of C for the root s of O.

We refer to (Jha and Suciu 2012, Lemma 4.1) for the correctness of the construction. Note also that (Jha and Suciu 2012, Lemma 4.1) provides a path decomposition $\mathcal{P}_O = (P_O, \lambda_O)$ of width at most $5\mathsf{width}(O)$ of C as follows. For a feature $f \in F(O)$, let $L(f)$ be the set of all vertices v of D with $\rho(v) = f$. Then, P_O has one vertex p_f for every but the last feature $f \in F(O)$ w.r.t. $<_O$. Moreover, let f be a feature in $F(O)$, whose successor w.r.t. $<_O$ is the feature f', then the bag $\lambda_O(p_f)$ is equal to $\{g_f, \overline{g_f}\} \cup \{ o_v \mid v \in L(f) \cup L(f') \} \cup \{ a_e \mid e = (u, v) \in E(D) \wedge u \in L(f) \wedge v \in L(f') \}$. This completes the description of $\mathcal{C}(O, c)$ for any $O \in \mathcal{O}$. It remains to show that $\mathcal{C}(\mathcal{O}, c)$ satisfies (2), in particular since the rankwidth is upper bounded by the pathwidth (using Lemma 1) it remains to show that the pathwidth of $\mathcal{C}(\mathcal{O}, c)$ is at most $|\mathcal{O}|5 \max_{O \in \mathcal{O}} \mathsf{width}(O)$. To see this consider the following path decomposition $\mathcal{P} = (P, \lambda)$ of $\mathcal{C}(\mathcal{O}, c)$. P has one vertex p_f for every but the last feature $f \in F(O)$ w.r.t. $<_O$. Moreover, $\lambda(p_f) = \{r\} \cup (\bigcup_{O \in \mathcal{O}} \lambda_O(p_f^O))$, where p_f^O is the vertex p_f of the path P_O of the path decomposition $\mathcal{P}_O = (P_O, \lambda_O)$. □

The following corollary follows immediately from Lemma 20 and Theorem 4.

Corollary 21. *Let* $\mathcal{P} \in \{\mathrm{LAX_P}, \mathrm{LCX_P}, \mathrm{GAX_P}, \mathrm{GCX_P}\}$. $\mathrm{OBDD}^<_{\mathrm{MAJ}}$-$\mathcal{P}_{||}(\mathit{ens\_size} + \mathit{width\_elem})$ *is* FPT.

The next Corollary follows again from Lemma 20 and Theorem 4 after observing that the pathwidth of the Boolean circuit constructed in Lemma 20 for an $\mathrm{OBDD}_{\mathrm{MAJ}}$ is bounded by $\mathcal{O}(\mathit{ens\_size} \cdot \mathit{size\_elem})$ since this bound is also an upperbound on the number of gates in the constructed Boolean circuit. Note that in contrast to $\mathrm{OBDD}^<_{\mathrm{MAJ}}$s, where the pathwidth of the resulting circuit can be bounded purely in terms of *ens_size* and *width_elem*, this is no longer the case for $\mathrm{OBDD}_{\mathrm{MAJ}}$s.

Corollary 22. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. $\text{OBDD}_{\text{MAJ}}\text{-}\mathcal{P}_{||}(\textit{ens\_size} + \textit{size\_elem})$ *is* FPT.

Theorem 23 ((Barceló et al. 2020, Lemma 14))**.** *There is a polynomial-time algorithm that given a* OBDD O *and an example* e *outputs a (cardinality-wise) minimum local contrastive explanation for* e *w.r.t.* O *or no if such an explanation does not exist. Therefore,* OBDD-$\text{LCXP}_{||}$ *can be solved in polynomial-time.*

The following auxiliary lemma is an analogue of Lemma 9 for OBDDs and provides polynomial-time algorithms for testing whether a given subset of features A/partial example e' is a local abductive/global abductive/global contrastive explanation for a given example e/class c w.r.t. a given OBDD O.

Lemma 24. *Let* O *be an* OBDD*, let* e *be an example and let* c *be a class. There are polynomial-time algorithms for the following problems:*

(1) Decide whether a given subset $A \subseteq F(O)$ *of features is a local abductive explanation for* e *w.r.t.* O*.*

(2) Decide whether a given partial example e' *is a global abductive explanation for* c *w.r.t.* O*.*

(3) Decide whether a given partial example e' *is a global contrastive explanation for* c *w.r.t.* O*.*

Proof. Let $O = (D, \rho)$ be an OBDD, let e be an example and let c be a class. For a partial example e', we denote by $D_{e'}$ the directed acylic graph obtained from D after removing all arcs from a vertex v whose feature $f = \rho(v)$ is assigned by e' to its $1 - e'(f)$-neighbor. For a subset $A \subseteq F(O)$ of features we denote by $e_{|A}$ the partial example equal to the restriction of e to A.

Towards showing (1), let $A \subseteq F(O)$ be a subset of features. Then, A is a local abductive explanation for e w.r.t. O if and only if $t_{O(e)}$ is the only sink vertex of $D_{e_{|A}}$ that is reachable from s, which can clearly be checked in polynomial-time.

Similarly, towards showing (2), observe that the partial example (assignment) $\tau : F \to \{0, 1\}$ is a global abductive explanation for c w.r.t. O if and only if t_c is the only sink reachable from s in D_τ, which can clearly be checked in polynomial-time.

Finally, towards showing (3), note that a partial example $\tau : F \to \{0, 1\}$ is a global contrastive explanation for c w.r.t. O if and only if t_c is not reachable from s in D_τ, which can clearly be verified in polynomial-time. □

Using dedicated algorithms for the inclusion-wise minimal variants of LAXP, GAXP, and GCXP together with Theorem 23, we obtain the following result.

Theorem 25. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. OBDD-$\mathcal{P}_{\subseteq}$ *can be solved in polynomial-time.*

Proof. Note that the statement of the theorem for OBDD-$\text{LCXP}_{\subseteq}$ follows immediately from Theorem 23. Therefore, it suffices to show the statement of the theorem for the remaining 3 problems.

Let (O, e) be an instance of OBDD-$\text{LAXP}_{\subseteq}$. We start by setting $A = F(O)$. Using Lemma 24, we then test for any feature f in A, whether $A \setminus \{f\}$ is still a local abductive explanation for e w.r.t. O in polynomial-time. If so, we repeat the process after setting A to $A \setminus \{f\}$ and otherwise we do the same test for the next feature $f \in A$. Finally, if $A \setminus \{f\}$ is not a local abductive explanation for every $f \in A$, then A is an inclusion-wise minimal local abductive explanation and we can output A.

The polynomial-time algorithm for a given instance (O, c) with $O = (D, \rho)$ for OBDD-$\text{GAXP}_{\subseteq}$ now works as follows. Let P be any shortest path from s to t_c in D; if no such path exists we can correctly output that there is no global abductive explanation for c w.r.t. O. Then, the assignment α defined by P is a global abductive explanation for c w.r.t. O. To obtain an inclusion-wise minimal solution, we do the following. Let $F = F(\alpha)$ be the set of features on which α is defined. We now test for every feature $f \in F$ whether the restriction $\alpha[F \setminus \{f\}]$ of α to $F \setminus \{f\}$ is a global abductive explanation for c w.r.t. $\mathcal{T}$. This can clearly be achieved in polynomial-time with the help of Lemma 24. If this is true for any feature $f \in F$, then we repeat the process for $\alpha[F \setminus \{f\}]$, otherwise we output α. A very similar algorithm now works for the DT-$\text{GCXP}_{\subseteq}$ problem, i.e., instead of starting with a shortest path from s to t_c, we start with a shortest path from s to $t_{c'}$ for some $c' \neq c$. □

Lemma 26. *Let* $\mathcal{O}$ *be an* $\text{OBDD}^{\leq}_{\text{MAJ}}$ *with* $\ell = |\mathcal{O}|$ *and* m *is the maximum size of any* OBDD *in* $\mathcal{O}$*. There is an algorithm that in time* $\mathcal{O}(m^\ell)$ *computes an* $\text{OBDD}^{<}$ O *of size at most* m^ℓ *such that* $O(e) = \mathcal{O}(e)$ *for every example* e*.*

Proof. We construct the OBDD $O = (D, \rho)$ as follows. Let D' be the directed acyclic graph obtained as follows. D' has one vertex n_p for every tuple $p = (v_1, \ldots, v_\ell) \in V(D_1) \times \cdots \times V(D_\ell)$. For a tuple $p = (v_1, \ldots, v_\ell) \in V(D_1) \times \cdots \times V(D_\ell)$, we denote by $\lambda(p)$ the smallest feature in $\{\, \rho_i(v_i) \mid 1 \leq i \leq \ell \,\}$ w.r.t. the ordering $<_{\mathcal{O}}$ if $\{\, \rho_i(v_i) \mid 1 \leq i \leq \ell \,\} \neq \emptyset$. Moreover, otherwise, i.e., if $\{\, \rho_i(v_i) \mid 1 \leq i \leq \ell \,\} = \emptyset$, then every vertex v_i is a sink vertex in D_i, i.e., it either corresponds to t_0 or to t_1, in which case we let $\lambda(p) = 1$ if the majority of the vertices $v_1, \ldots, v_\ell$ correspond to t_1 in D_i and otherwise we set $\lambda(p) = 0$. Let p be such that $\lambda(p) \neq \emptyset$ and let $b \in \{0, 1\}$. The b-successor of p, denoted by $S_b(p)$, is the tuple $(v'_1, \ldots, v'_k)$, where $v'_i = v_i$ if $\rho_i(v_i) \neq \lambda(p)$ and v'_i is the b-neighbor of v_i in D_i, otherwise. Now, every vertex n_p in D' with $\lambda(p) \neq \emptyset$ has $n_{S_0(p)}$ as its 0-neighbor and $n_{S_1(p)}$ as its 1-neighbor and this completes the description of D'. Note that D' is acylic because it only contains arcs from a vertex n_p with $\lambda(p) \notin \{0, 1\}$ to a vertex $n_{(S_b(p)}$ and either $\lambda(n_p) <_{\mathcal{O}} \lambda(S_b(p))$ or $\lambda(S_b(p)) \in \{0, 1\}$.

Then, D is the directed acyclic graph with source vertex $n_{(s_0,\ldots,s_\ell)}$ and sink vertices t_0 and t_1 that is obtained from D' after applying the following operations:

- Remove all vertices from D' that are not reachable from the vertex $n_{(s_0,\ldots,s_\ell)}$, where s_i is the root of D_i.
- Identify the vertices in $\{\, n_p \mid \lambda(p) = b \,\}$ with the new vertex t_b for every $b \in \{0, 1\}$.

Finally, we set $\rho(n_p) = \lambda(p)$ for every p with $\lambda(p) \notin \{0, 1\}$. This completes the construction of O, which can clearly be achieved in time $\mathcal{O}(m^\ell)$ and has size at most m^ℓ. It is now straightforward to verify that $\mathcal{O}(e) = O(e)$ for every example, as required. □

The next theorem uses our result that the considered problems are in polynomial-time for OBDDs together with an XP-algorithm that transforms any $\text{OBDD}^<_{\text{MAJ}}$ into an equivalent OBDD.

Theorem 27. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{LCXP}, \text{GAXP}, \text{GCXP}\}$. $\text{OBDD}^<_{\text{MAJ}}$-$\mathcal{P}_\subseteq$(*ens_size*) *and* $\text{OBDD}^<_{\text{MAJ}}$-$\text{LCXP}_{|\cdot|}$(*ens_size*) *are in* XP.

Proof. Let $\mathcal{O} = \{O^1, \ldots, O^\ell\}$ with variable ordering $<_\mathcal{O}$ and $O_i = (D_i, \rho_i)$ be the $\text{OBDD}^<_{\text{MAJ}}$ given as an input to any of the five problems stated above.

We use Lemma 26 to construct an OBDD O that is not too large (of size at most m^ℓ, where $m = (\max_{i=1}^{\ell} |V(D_i)|)$) and that is equivalent to $\mathcal{O}$ in the sense that $\mathcal{O}(e) = O(e)$ for every example in time at most $\mathcal{O}(m^\ell)$. Since all five problems can be solved in polynomial-time on O (because of Theorem 23 and Theorem 25) this completes the proof of the theorem.

□

6 Hardness Results

In this section, we provide our algorithmic lower bounds. We start by showing a close connection between the complexity of all of our explanation problems to the following two problems. As we will see the hardness of finding explanations comes from the hardness of deciding whether or not a given model classifies all examples in the same manner. More specifically, from the HOM problem defined below, which asks whether a given model has an example that is classified differently from the all-zero example, i.e., the example being 0 on every feature. We also need the P-HOM problem, which is a parameterized version of HOM that we use to show parameterized hardness results for deciding the existence of local contrastive explanations.

In the following, let $\mathcal{M}$ be a model type.

$\mathcal{M}$-HOMOGENEOUS (HOM)	
INSTANCE:	A model $M \in \mathcal{M}$.
QUESTION:	Is there an example e such that $M(e) \neq M(e_0)$, where e_0 is the all-zero example?

$\mathcal{M}$-P-HOMOGENEOUS (P-HOM)	
INSTANCE:	A model $M \in \mathcal{M}$ and integer k.
QUESTION:	Is there an example e that sets at most k features to 1 such that $M(e) \neq M(e_0)$, where e_0 is the all-zero example?

The following lemma now shows the connection between HOM and the considered explanation problems.

Lemma 28. *Let $M \in \mathcal{M}$ be a model, e_0 be the all-zero example, and let $c = M(e_0)$. The following problems are equivalent:*

(1) M is a no-instance of $\mathcal{M}$-HOM.

(2) The empty set is a solution for the instance (M, e_0) of $\mathcal{M}$-$\text{LAXP}_\subseteq$.

(3) (M, e_0) is a no-instance of $\mathcal{M}$-$\text{LCXP}_\subseteq$.

(4) The empty set is a solution for the instance (M, c) of $\mathcal{M}$-$\text{GAXP}_\subseteq$.

(5) The empty set is a solution for the instance $(M, 1-c)$ of $\mathcal{M}$-$\text{GCXP}_\subseteq$.

(6) $(M, e_0, 0)$ is a yes-instance of $\mathcal{M}$-$\text{LAXP}_{|\cdot|}$.

(7) (M, e_0) is a no-instance of $\mathcal{M}$-$\text{LCXP}_{|\cdot|}$.

(8) $(M, c, 0)$ is a yes-instance of $\mathcal{M}$-$\text{GAXP}_{|\cdot|}$.

(9) $(M, 1-c, 0)$ is a yes-instance of $\mathcal{M}$-$\text{GCXP}_{|\cdot|}$.

Proof. It is easy to verify that all of the statements (1)–(9) are equivalent to the following statement (and therefore equivalent to each other): $M(e) = M(e_0) = c$ for every example e. □

While Lemma 28 is sufficient for most of our hardness results, we also need the following lemma to show certain parameterized hardness results for deciding the existence of local contrastive explanations.

Lemma 29. *Let $M \in \mathcal{M}$ be a model and let e_0 be the all-zero example. The following problems are equivalent:*

(1) (M, k) is a yes-instance of $\mathcal{M}$-P-HOM.

(2) (M, e_0, k) is a yes-instance of $\mathcal{M}$-$\text{LCXP}_{|\cdot|}$.

Proof. The lemma follows because both statements are equivalent to the following statement: There is an example e that sets at most k features to 1 such that $M(e) \neq M(e_0)$. □

We will often reduce from the following problem, which is well-known to be NP-hard and also W[1]-hard parameterized by k.

MULTICOLORED CLIQUE (MCC)	
INSTANCE:	A graph G with a proper k-coloring of $V(G)$.
QUESTION:	Is there a clique of size k in G?

The following lemma provides a unified way to show hardness results for ensembles for practically all of our model types in the case that we allow arbitrarily many (constant-size) ensemble elements, i.e., we use it to show Theorems 36, 40, 44.

Lemma 30. *Let $\mathcal{M}$ be a class of models such that there are models $M^0 \in \mathcal{M}$, $M^1_f \in \mathcal{M}$ and $M^2_{f_1,f_2} \in \mathcal{M}$ for features f, f_1, and f_2 of size at most d such that:*

- *M^0 classifies every example negatively.*
- *M^1_f classifies an example e positively iff $e(f) = 1$.*
- *$M^2_{f_1,f_2}$ classifies an example e positively iff $e(f_1) = 0$ or $e(f_2) = 0$.*

$\mathcal{M}_{\text{MAJ}}$-P-HOM is W[1]-hard parameterized by k even if the size of each ensemble element is at most d and $\mathcal{M}_{\text{MAJ}}$-HOM is NP-hard even if the size of each ensemble element is at most d.

Proof. We provide a parameterized reduction from the MULTICOLORED CLIQUE (MCC) problem, which is also a polynomial-time reduction. Given an instance (G, k) of the MCC problem with k-partition $(V_1, \ldots, V_k)$ of $V(G)$, we will construct an equivalent instance $(\mathcal{E}, k)$ of $\mathcal{M}_{\text{MAJ-P-HOM}}$ in polynomial-time as follows. $\mathcal{E}$ uses one binary feature f_v for every $v \in V(G)$. Let $<_V$ be an arbitrary ordering of $V(G)$. We denote by n and m the number of vertices and edges of the graph G, respectively.

$\mathcal{E}$ contains the following ensemble elements:

- For every non-edge $uv \notin E(G)$ with $u <_V v$, we add the model $M^2_{f_u,f_v}$ to $\mathcal{E}$.
- For every vertex $v \in V(G)$, we add the model $M^1_{f_v}$ to $\mathcal{E}$.
- We add $(\binom{n}{2} - m) - n + 2k - 1$ models M^0 to $\mathcal{E}$.

Clearly, the reduction works in polynomial-time and preserves the parameter and it only remains to show that G has a k-clique if and only if there is an example e such that $\mathcal{E}(e) \neq \mathcal{E}(e_0)$ that sets at most k features to 1. Note first that $\mathcal{E}(e_0) = 0$ because $M(e_0) = 1$ holds only for $(\binom{n}{2} - m)$ out of the $2(\binom{n}{2} - m) + 2k - 1$ models in $\mathcal{E}$.

Towards showing the forward direction, let $C = \{v_1, \ldots, v_k\}$ be a k-clique of G with $v_i \in V_i$ for every $i \in [k]$. We claim that the example e that sets all features in $\{ f_v \mid v \in C \}$ to 1 and all other features to 0 satisfies $\mathcal{E}(e) \neq \mathcal{E}(e_0)$. Because C is a clique in G, we obtain that $M(e) = 1$ for all of the $(\binom{n}{2} - m)$ copies M of $M^2_{f_u,f_v}$ in $\mathcal{E}$. Moreover, because e sets exactly k features to 1, it holds that $M(e) = 1$ for exactly k copies M of $M^1_{f_v}$ in $\mathcal{E}$. Therefore, e is classified positively by exactly $(\binom{n}{2} - m) + k$ models in $\mathcal{E}$ and e is classified negatively by exactly $n - k + (\binom{n}{2} - m) - n + 2k - 1 = (\binom{n}{2} - m) + k - 1$ models in $\mathcal{E}$, which shows that $\mathcal{E}(e) = 1 \neq \mathcal{E}(e_0)$.

Towards showing the reverse direction, let e be an example that sets at most k features to 1 such that $\mathcal{E}(e) = 1 \neq \mathcal{E}(e_0)$. First note that because $M(e) = 0$ for every of the $(\binom{n}{2} - m) - n + 2k - 1$ copies M of M^0 in $\mathcal{E}$, it has to hold that $M(e) = 1$ for at least $(\binom{n}{2} - m) + k$ of the copies M of either $M^1_{f_v}$ or $M^2_{f_u,f_v}$ in $\mathcal{E}$. Since e sets only at most k features to 1, we obtain that $M(e) = 1$ for at most k of the copies M of $M^1_{f_v}$ in $\mathcal{E}$. Therefore, $M(e) = 1$ for all copies of M of $M^2_{f_u,f_v}$ in $\mathcal{E}$ and moreover $M(e) = 1$ for exactly k copies M of $M^1_{f_v}$ in $\mathcal{E}$. But then the set $\{ v \mid e(f_v) = 1 \}$ must be a k-clique of G. □

6.1 DTs and their Ensembles

Here, we provide our algorithmic lower bounds for DTs and their ensembles. We say that a DT $\mathcal{T}$ is *ordered* if there is an ordering $<$ of the features in $F(\mathcal{T})$ such that the ordering of the features on every root-to-leaf path of $\mathcal{T}$ agrees with $<$. We need the following auxiliary lemma to simplify the descriptions of our reductions.

Lemma 31. *Let $E \subseteq E(F)$ be a set of examples defined on features in F. An ordered* DT *$\mathcal{T}_E$ of size at most $2|E||F|+1$ such that $\mathcal{T}_E(e) = 1$ if and only if $e \in E$ can be constructed in time $\mathcal{O}(|E||F|)$.*

Proof. Let $< = (f_1, \ldots, f_n)$ be an arbitrary order of the features in F. First, we construct a simple ordered DT $\mathcal{T}_e = (T_e, \lambda_e)$ that classifies only example e as 1 and all other examples as 0. $\mathcal{T}_e$ has one inner node t^e_i for every $i \in [n]$ with $\lambda_e(t^e_i) = f_i$. Moreover, for $i < n$, t^e_i has t^e_{i+1} as its $e(f_i)$-child and a new 0-leaf as its other child. Finally, t^e_n has a new 1-leaf as its $e(f_n)$-child and a 0-leaf as its other child. Clearly, $\mathcal{T}_e$ can be constructed in time $\mathcal{O}(|F|)$.

We now construct $\mathcal{T}_E$ iteratively starting from $\mathcal{T}_\emptyset$ and adding one example from E at a time (in an arbitrary order). We set $\mathcal{T}_\emptyset$ to be the DT that only consists of a 0-leaf. Now to obtain $\mathcal{T}_{E' \cup \{e\}}$ from $\mathcal{T}_{E'}$ for some $E' \subseteq E$ and $e \in E \setminus E'$, we do the following. Let l be the 0-leaf of $\mathcal{T}_{E'}$ that classifies e and let f_i be the feature assigned to the parent of l. Moreover, let $\mathcal{T}'_e$ be the sub-DT of $\mathcal{T}_e$ rooted at t^e_{i+1} or if $i = n$ let $\mathcal{T}'_e$ be the DT consisting only of a 1-leaf. Then, $\mathcal{T}_{E' \cup \{e\}}$ is obtained from the disjoint union of $\mathcal{T}_{E'}$ and $\mathcal{T}'_e$ after identifying the root of $\mathcal{T}'_e$ with l. Clearly, $\mathcal{T}_E$ is an ordered DT that can be constructed in time $\mathcal{O}(|E||F|)$ has size at most $2|E||F| + 1$ and satisfies $\mathcal{T}_E(e) = 1$ if and only if $e \in E$. □

We note that the following theorem also follows from a result in (Barceló et al. 2020, Proposition 5) for FBDDs, i.e., BDDs without contradicting paths. However, we require a different version of the proof that generalizes easily to OBDDs, i.e., we need to show hardness for ordered DTs.

Theorem 32. DT-LAX$_{\text{PI}}$ *is* NP*-hard and* DT-LAX$_{\text{PI}}$(*xp_size*) *is* W[2]*-hard even if for ordered* DT*s.*

Proof. We provide a parameterized reduction from the HITTING SET problem, which is well-known to be NP-hard and W[2]-hard parameterized by the size of the solution. That is, given a family of sets $\mathcal{F}$ over a universe U and an integer k, the HITTING SET problem is to decide whether $\mathcal{F}$ has a *hitting set* of size at most k, i.e., a subset $S \subseteq U$ with $|U| \leq k$ such that $S \cap F \neq \emptyset$ for every $F \in \mathcal{F}$. Given an instance $(U, \mathcal{F}, k)$ of the HITTING SET problem, we will construct an equivalent instance of $(\mathcal{T}, e, k)$ LAX$_{\text{PI}}$ in polynomial-time as follows.

Let B be the set of features containing one binary feature f_u for every $u \in U$ and let E be the set of examples equal to $\{ e_F \mid F \in \mathcal{F} \}$, where e_F is the example that is 1 at every feature f_u such that $u \in F$ and otherwise 0. Using Lemma 31 we can construct an ordered DT $\mathcal{T}_E$ of size at most $2|E||B|$ such that $\mathcal{T}_E(e) = 1$ if and only if $e \in E$. Let $\mathcal{T} = \mathcal{T}_E$ and let e be the all-zero example. Clearly, $(\mathcal{T}, e, k)$ can be constructed from $(U, \mathcal{F}, k)$ in polynomial-time and it only remains to show the equivalence of the two instances.

Towards showing the forward direction, let S be a hitting set for $\mathcal{F}$ of size at most k. We claim that $A = \{ f_u \mid u \in S \}$ is a local abductive explanation for e w.r.t. $\mathcal{T}$, which concludes the proof of the forward direction. To see that this is indeed the case consider any example e' that agrees with e on A, i.e., e' is 0 at every feature in A. Then, because S is a hitting set for $\mathcal{F}$, we obtain that e' is not in E, which implies that $\mathcal{T}(e') = 0 = \mathcal{T}(e)$, as required.

Towards showing the reverse direction, let A be a local abductive explanation of size at most k for e w.r.t. $\mathcal{T}$. We claim that $S = \{ u \mid f_u \in A \}$ is a hitting set for $\mathcal{F}$. Suppose that this is not the case and there is a set $F \in \mathcal{F}$ with $F \cap S = \emptyset$. Then, the example e_F agrees with e on every feature in A, however, it holds that $\mathcal{T}(e_F) = 1 \neq \mathcal{T}(e)$, contradicting our assumption that A is a local abductive explanation for e w.r.t. $\mathcal{T}$. □

The following theorem is an analogue of Theorem 32 for global abductive and global contrastive explanations. It is interesting to note that while it was not necessary to distinguish between local abductive explanations on one side and global abductive and global contrastive explanations on the other side in the setting of algorithms, this is no longer the case when it comes to algorithmic lower bounds. Moreover, while the following result establishes W[1]-hardness for DT-GAXP$_{||}$(*xp_size*) and DT-GCXP$_{||}$(*xp_size*), this is achieved via fpt-reductions that are not polynomial-time reductions, which is a behavior that is very rarely seen in natural parameterized problems. While it is therefore not clear whether the problems are NP-hard, the result still shows that the problems are not solvable in polynomial-time unless FPT = W[1], which is considered unlikely (Downey and Fellows 2013).

Theorem 33. DT-GAXP$_{||}$(*xp_size*) *and* DT-GCXP$_{||}$(*xp_size*) *are* W[1]*-hard. Moreover, there is no polynomial time algorithm for solving* DT-GAXP$_{||}$ *and* DT-GCXP$_{||}$*, unless* FPT = W[1].

Proof. We provide a parameterized reduction from the MULTICOLORED CLIQUE (MCC) problem, which is well-known to be W[1]-hard parameterized by the size of the solution. Given an instance (G, k) of the MCC problem with k-partition $(V_1, \ldots, V_k)$ of $V(G)$, we will construct an equivalent instance $(\mathcal{T}, c, k)$ of GAXP$_{||}$ in fpt-time. Note that since a partial example e' is a global abductive explanation for c w.r.t. $\mathcal{T}$ if and only if e' is a global contrastive explanation for $1 - c$ w.r.t. $\mathcal{T}$, this then also implies the statement of the theorem for GCXP$_{||}$. $\mathcal{T}$ uses one binary feature f_v for every $v \in V(G)$.

We start by constructing the DT $\mathcal{T}_{i,j}$ for every $i, j \in [k]$ with $i \neq j$ satisfying the following: (*) $\mathcal{T}_{i,j}(e) = 1$ for an example e if and only if either $e(f_v) = 0$ for every $v \in V_i$ or there exists $v \in V_i$ such that $e(f_v) = 1$ and $e(f_{v'}) = 0$ for every $v' \in (V_i \setminus \{v\}) \cup (N_G(v) \cap V_j)$. Let $\mathcal{T}_i$ be the DT obtained using Lemma 31 for the set of examples $\{e_0\} \cup \{ e_v \mid v \in V_i \}$ defined on the features in $F_i = \{ f_v \mid v \in V_i \}$. Here, e_0 is the all-zero example and for every $v \in V_i$, e_v is the example that is 1 only at the feature f_v and 0 otherwise. Moreover, for every $v \in V_i$, let $\mathcal{T}_j^v$ be the DT obtained using Lemma 31 for the set of examples containing only the all-zero example defined on the features in $\{ f_{v'} \mid v' \in N_G(v) \cap V_j \}$. Then, $\mathcal{T}_{i,j}$ is obtained from $\mathcal{T}_i$ after replacing the 1-leaf that classifies e_v with $\mathcal{T}_j^v$ for every $v \in V_i$. Clearly, $\mathcal{T}_{i,j}$ satisfies (*) and since $\mathcal{T}_i$ has at most $|V_i|^2$ inner nodes and $\mathcal{T}_j^v$ has at most $|V_j|$ inner nodes, we obtain that $\mathcal{T}_{i,j}$ has at most $\mathcal{O}(|V(G)|^2)$ nodes.

For an integer ℓ, we denote by DT(ℓ) the complete DT of height ℓ, where every inner node is assigned to a fresh auxiliary feature and every of the exactly 2^ℓ leaves is a 0-leaf. Let $\mathcal{T}_\Delta$ be the DT obtained from the disjoint union of $\mathcal{T}_U = \text{DL}(k)$ and 2^k copies $\mathcal{T}_D^1, \ldots, \mathcal{T}_D^{2^k}$ of DT ($\lceil \log(k(k-1)) \rceil$) by identifying the i-th leaf of $\mathcal{T}_U$ with the root of $\mathcal{T}_D^i$ for every i with $1 \leq i \leq 2^k$; each copy is equipped with its own set of fresh features.

Then, $\mathcal{T}$ is obtained from $\mathcal{T}_\Delta$ after doing the following with $\mathcal{T}_D^\ell$ for every $\ell \in [2^k]$. For every $i, j \in [k]$ with $i \neq j$, we replace a private leaf of $\mathcal{T}_D^\ell$ with the DT $\mathcal{T}_{i,j}$; note that this is possible because $\mathcal{T}_D^\ell$ has at least $k(k-1)$ leaves. Also note that $\mathcal{T}$ has size at most $\mathcal{O}(|\mathcal{T}_\Delta||V(G)|^2)$. This completes the construction of $\mathcal{T}$ and we set $c = 0$. Clearly, $\mathcal{T}$ can be constructed from G in fpt-time w.r.t. k. It remains to show that G has a k-clique if and only if there is a global abductive explanation of size at most k for c w.r.t. $\mathcal{T}$.

Towards showing the forward direction, let $C = \{v_1, \ldots, v_k\}$ be a k-clique of G with $v_i \in V_i$ for every $i \in [k]$. We claim that $\alpha : \{ f_v \mid v \in V(C) \} \to \{1\}$ is a global abductive explanation for c w.r.t. $\mathcal{T}$, which concludes the proof of the forward direction. To see that this is indeed the case consider any example e that agrees with α, i.e., e is 1 at any feature f_v with $v \in V(C)$. For this is suffices to show that $\mathcal{T}_{i,j}(e) = 0$ for every $i, j \in [k]$. Because $\mathcal{T}_{i,j}$ satisfies (*) and because $e(f_{v_i}) = 1$, it has to hold that e is 1 for at least one feature in $\{ f_v \mid v \in N_G(v_i) \cap V_j \}$. But this clearly holds because C is a k-clique in G.

Towards showing the reverse direction, let $\alpha : C' \to \{0, 1\}$ be a global abductive explanation of size at most k for c w.r.t. $\mathcal{T}$. We claim that $C = \{ v \mid f_v \in C' \}$ is a k-clique of G. Let $\mathcal{T}_\alpha$ be the partial DT obtained from $\mathcal{T}$ after removing all nodes that can never be reached by an example that is compatible with α, i.e., we obtain $\mathcal{T}_\alpha$ from $\mathcal{T}$ by removing the subtree rooted at the $1 - \alpha(t)$-child for every node t of $\mathcal{T}$ with $\lambda(t) \in C'$. Then, $\mathcal{T}_\alpha$ contains only 0-leaves. We first show that there is an $\ell \in [2^k]$ such that $\mathcal{T}_\alpha$ contains $\mathcal{T}_D^\ell$ completely, i.e., this in particular means that C' contains no feature of $\mathcal{T}_D^\ell$. To see this, let x be the number of features in C' that are assigned to a node of $\mathcal{T}_U$. Then, $\mathcal{T}_\alpha$ contains the root of at least 2^{k-x} DTs $\mathcal{T}_D^i$. Moreover, since $2^{k-x} \geq k - x$ there is at least one $\mathcal{T}_D^i$ say $\mathcal{T}_D^\ell$, whose associated features are not in C'. Therefore, for every $i, j \in [k]$ with $i \neq j$, $\mathcal{T}_\alpha$ contains at least the root of $\mathcal{T}_{i,j}$. Since $\mathcal{T}_{i,j}$ satisfies (*), it follows that for every $\ell \in [k]$ there is $v_\ell \in V_\ell$ such that $\alpha(f_{v_\ell}) = 1$. Because $|C'| \leq k$, we obtain that C' contains exactly one feature $f_{v_\ell} \in F_\ell$ for every ℓ. Finally, using again that $\mathcal{T}_{i,j}$ satisfies (*), we obtain that v_i and v_j are adjacent in G, showing that $\{v_1, \ldots, v_k\}$ is a k-clique of G. □

Lemma 34. DT$_{\text{MAJ}}$-HOM *is* NP*-hard and both* DT$_{\text{MAJ}}$-HOM(*ens_size*) *and* DT$_{\text{MAJ}}$-P-HOM(*ens_size*) *are* W[1]*-hardeven if the partial orders of the features on all paths from the root to a leaf respect the same total order.*

Proof. We give a parameterized reduction from MCC that is also a polynomial-time reduction. That is, given an instance (G, k) of MCC with k-partition $V_1, \ldots, V_k$, we will

construct a DT_{MAJ} $\mathcal{E}$ with $|\mathcal{E}| = 2(k + \binom{k}{2}) - 1$ such that G has a k-clique if and only if $\mathcal{E}$ classifies at least one example positively. This will already suffice to show the stated results for DT_{MAJ}-HOM. Moreover, to show the results for DT_{MAJ}-P-HOM we additional show that G has a k-clique if and only if $\mathcal{E}$ classifies an example positively that sets at most k features to 1.

$\mathcal{E}$ will use the set of features $\bigcup_{i\in[k]} F_i$, where $F_i = \{ f_v \mid v \in V_i \}$. For each $v \in V_i$ and $u \in V_j$, let $e_{v,u}$ be an example defined on set of features $F_i \cup F_j$ that is 1 only at the features f_v and f_v, and otherwise 0. For every $i \in [k]$, $\mathcal{E}$ will have a DT $\mathcal{T}_i$ obtained using Lemma 31 for the set of examples $\{ e_{v,v} \mid v \in V_i \}$ defined on the features in F_i. Also, for every i and j with $1 \leq i < j \leq k$, $\mathcal{E}$ contains a DT $\mathcal{T}_{i,j}$ obtained using Lemma 31 for the set of examples $\{ e_{v,u} \mid v \in V_i \wedge u \in V_j \wedge vu \in E(G) \}$ defined on the features in $F_i \cup F_j$. Finally, $\mathcal{E}$ contains $k + \binom{k}{2} - 1$ DTs that classify every example negatively, i.e., those DTs consists only of one 0-leaf. Clearly, the reduction can be achieved in polynomial-time and preserves the parameter, i.e., $|\mathcal{E}| = 2(k + \binom{k}{2}) - 1$. It remains to show that G has a k-clique if and only if $\mathcal{E}$ classifies at least one example positively, which in turn holds if and only if $\mathcal{E}$ classifies an example positively that sets at most k features to 1.

Towards showing the forward direction, let $C = \{ v_1, \ldots, v_k \}$ be a k-clique of G, where $v_i \in V_i$ for every i with $1 \leq i \leq k$. We claim that the example e that is 1 exactly at the features $f_{v_1}, \ldots, f_{v_k}$ (and otherwise 0) is classified positively by $\mathcal{E}$. By construction, e is classified positively by every DT $\mathcal{T}_i$ for every $i \in [k]$ and since C is a k-clique also every DT $\mathcal{T}_{i,j}$ for every $1 \leq i < j \leq k$. Therefore, e is classified positively by $k + \binom{k}{2}$ DT, which represents the majority of the DTs in $\mathcal{E}$.

Towards showing the reverse direction, suppose that there is an example e that is classified positively by $\mathcal{E}$. Because e has to be classified positively by the majority of DT in $\mathcal{E}$ and there are $k + \binom{k}{2} - 1$ DTs in $\mathcal{E}$ that classify e negatively, we obtain that e has to be classified positively by every DT $\mathcal{T}_i$ for every $i \in [k]$ and by every DT $\mathcal{T}_{i,j}$ for every $1 \leq i < j \leq k$. Since e is classified positively by $\mathcal{T}_i$, we obtain by construction that there is exactly one feature $f_{v_i} \in \{ f_v \mid v \in V_i \}$ such that $e(f_{v_i}) = 1$ and e is 0 at all other features in $f_{v_i} \in \{ f_v \mid v \in V_i \}$. Since e sets exactly one feature $f_{v_i} \in \{ f_v \mid v \in V_i \}$ to 1 and e sets exactly one feature $f_{v_j} \in \{ f_v \mid v \in V_i \}$ to 1 and because e is classified positively by $\mathcal{T}_{i,j}$, we obtain from the construction that v_i and v_j are adjacent in G. Therefore, $C = \{v_1, \ldots, v_k\}$ is a k-clique of G. □

The final two theorems of this section provide all the remaining hardness results for DT_{MAJ}s and follow from Lemma 34 and Lemma 30, respectively, together with Lemmas 28, 29.

Theorem 35. *Let $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. DT_{MAJ}-$\mathcal{P}_{\subseteq}$(ens_size) is* co-W[1]*-hard; DT_{MAJ}-$\text{LCXP}_{\subseteq}$(ens_size) is* W[1]*-hard; DT_{MAJ}-$\mathcal{P}_{||}$(ens_size) is* co-W[1]*-hard even if xp_size is constant; DT_{MAJ}-$\text{LCXP}_{||}$(ens_size + xp_size) is* W[1]*-hard.*

Theorem 36. *Let $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. DT_{MAJ}-$\mathcal{P}_{\subseteq}$ is* co-NP*-hard even if mnl_size + size_elem is constant; DT_{MAJ}-$\text{LCXP}_{\subseteq}$ is* NP*-hard even if mnl_size + size_elem is constant; DT_{MAJ}-$\mathcal{P}_{||}$ is* co-NP*-hard even if mnl_size + size_elem + xp_size is constant; DT_{MAJ}-$\text{LCXP}_{||}$(xp_size) is* W[1]*-hard even if mnl_size + size_elem is constant.*

Proof. We will use Lemma 30 to show the theorem by giving DTs M^0, M^1_f and $M^2_{\{f_1,f_2\}}$ satisfying the conditions of Lemma 30 as follows. We let M^0 be the DT consisting merely of one 0-leaf. We let M^1_f be the DT (T^1_f, λ^1_f), such that t is the only inner node of T^1_f with $\lambda^1_f(t) = f$. Let 0-leaf and 1-leaf be the 0-child and 1-child of t, respectively.

We let $M^2_{\{f_1,f_2\}}$ be the DT $(T^2_{\{f_1,f_2\}}, \lambda^2_{\{f_1,f_2\}})$, such that t^{f_1} and t^{f_2} are the only inner nodes of $T^2_{\{f_1,f_2\}}$ with $\lambda^2_{\{f_1,f_2\}}(t^{f_1}) = f_1$ and $\lambda^2_{\{f_1,f_2\}}(t^{f_2}) = f_2$. Let 1-leaf and t^{f_2} be the 0-child and 1-child of t^{f_1}, respectively. Let 1-leaf and 0-leaf be the 0-child and 1-child of g^{f_2}, respectively.

The constructions of the DTs are self explainable. Note that M^0, M^1_f and $M^2_{\{f_1,f_2\}}$ have *size_elem* at most 5. All statements in the theorem now follow from Lemmas 28 to 30. □

6.2 DSs, DLs and their Ensembles

Here, we establish our hardness results for DS, DLs, and their ensembles. It is interesting to note that there is no real distinction between DS and DLs when it comes to explainability and that both are considerably harder to explain than DTs and OBDDs.

3-DNF TAUTOLOGY (TAUT)
INSTANCE: A 3-DNF formula ψ.
QUESTION: Is there an assignment that falsifies ψ.

Theorem 37. *Let $\mathcal{M} \in \{\text{DS}, \text{DL}\}$ and let $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. $\mathcal{M}$-$\mathcal{P}_{\subseteq}$ is* co-NP*-hard even if term_size is constant; $\mathcal{M}$-$\text{LCXP}_{\subseteq}$ is* NP*-hard even if term_size is constant; $\mathcal{M}$-$\mathcal{P}_{||}$ is* co-NP*-hard even if term_size + xp_size is constant.*

Proof. Because every DS can be easily transformed into an equivalent DL that shares all of the same parameters, it is sufficient to show the theorem for the case that $\mathcal{M} = \text{DS}$.

First, we give a polynomial-time reduction from TAUT to DS-HOM. Let ψ be the 3-DNF formula given as an instance of TAUT. W.l.o.g., we can assume that the all-zero assignment satisfies ψ, since otherwise ψ is a trivial no-instance. Then, the DS $S = (T_\psi, 0)$, where T_ψ is the set of terms of ψ is the constructed instance of DS-HOM. Since the reduction is clearly polynomial, it only remains to show that ϕ is a tautology if and only if S classifies all examples in the same manner as the all-zero example, i.e., positively. But this is easily seen to hold, which concludes the reduction from TAUT to DS-HOM. All statements in the theorem now follow from Lemma 28. □

Theorem 38. *Let $\mathcal{M} \in \{\text{DS}, \text{DL}\}$. $\mathcal{M}$-$\text{LCXP}_{||}$(xp_size) is* W[1]*-hard.*

Proof. Again it suffices to show the theorem for the case that $\mathcal{M} = \text{DS}$. We provide a parameterized reduction from MCC to DS-P-HOM, which is also a polynomial-time reduction. Given an instance (G, k) of MCC with a k-partition $V_1, \ldots, V_k$, we will construct a DS $S = (T_{(1)} \cup T_{(2)}, 1)$ such that G has a k-clique if and only if S classifies at least one example positively, that sets at most k features to 1. S will use one binary feature f_v for every vertex $v \in V(G)$ and will classify an example e positively if and only if all of the following items are true:

(1) for every non-edge $uv \notin E(G)$, either the feature f_u or the feature f_v are 0 in e, i.e., $e(f_u) = 0$ or $e(f_v) = 0$
(2) for every $i \in [k]$, at least one of the features in $\{ f_v \mid v \in V_i \}$ is 1 in e,

For every property (i) from the list above, we create a set of terms $T_{(i)}$ such that an example e does not satisfy any term in $T_{(i)}$ if and only if it satisfies (i). Let $T_{(1)}$ be the set of terms equal to $\{ \{(f_u = 1), (f_v = 1)\} \mid uv \notin E(G) \}$. For every $1 \leq i \leq k$, let t_i be the term $\{ (f_v = 0) \mid v \in V_i \}$. Let $T_{(2)}$ be the set of terms equal to $\{ t_i \mid 1 \leq i \leq k \}$.

Clearly, the reduction can be achieved in polynomial-time. It remains to show that G has a k-clique if and only if S classifies at least one example positively, that sets at most k features to 1. Towards showing the forward direction, let $C = \{ v_1, \ldots, v_k \}$ be a k-clique of G. We claim that the example e that is 1 exactly at the features $\{ f_{v_1}, \ldots f_{v_k} \}$ (and otherwise 0) is classified positively by S. By construction, e does not satisfy any of the terms from $T_{(2)}$. Moreover, since C is a k-clique also no term from $T_{(1)}$ is satisfied by e. Therefore, e is classified positively by S.

Towards showing the reverse direction, suppose that there is an example e, that sets at most k features to 1, which is classified positively by S, i.e., no term from $T_{(1)}$ and $T_{(2)}$ is satisfied by e. Since e does not satisfy $T_{(2)}$ we get that e sets at least k to 1, i.e., one feature f_v with $v \in V_i$ for every $i \in [k]$. Moreover, since e does not satisfy $T_{(1)}$, all the vertices v with $e(f_v) = 1$ form a clique in G. Therefore, $C = \{v | e(f_v) = 1 \wedge v \in V(G)\}$ is a k-clique of G.

Note that S classifies the example e_0 negatively. This implies that DS-P-HOM is W[1]-hard and because of Lemma 29 we obtain that DS-LCXP$_{||}$(*xp_size*) is W[1]-hard. □

Theorem 39. *Let* $\mathcal{M} \in \{\text{DS}_{\text{MAJ}}, \text{DL}_{\text{MAJ}}\}$. $\mathcal{M}$-LCXP$_{||}$(*ens_size* + *xp_size*) *is* W[1]*-hard even if term_size is constant.*

Proof. Again it suffices to show the theorem for the case that $\mathcal{M} = \text{DS}_{\text{MAJ}}$. We provide a parameterized reduction from MCC to DS$_{\text{MAJ}}$-P-HOM, which is also a polynomial-time reduction. Given an instance (G, k) of MCC with a k-partition $V_1, \ldots, V_k$, we will construct a DS$_{\text{MAJ}}$ $\mathcal{S}$ such that $|\mathcal{S}| = 2k + 1$, *term_size* is equal to 2 and G has a k-clique if and only if $\mathcal{S}$ classifies at least one example positively, that sets at most k features to 1. We will construct DS$_{\text{MAJ}}$ $\mathcal{S} = \{ S_{(1)}, S^1_{(2)}, \ldots, S^k_{(2)}, S^1_{\perp}, \ldots, S^k_{\perp} \}$ which use one binary features f_v for every vertex $v \in V(G)$ and will classify an example e positively if and only if all of the following items are true:

(1) for every non-edge $uv \notin E(G)$, either the feature f_u or the feature f_v are 0 in e, i.e., $e(f_u) = 0$ or $e(f_v) = 0$
(2) for every $i \in [k]$, at least one of the features in $\{ f_v \mid v \in V_i \}$ is 1 in e,

Let $T_{(1)}$ be the set of terms equal to $\{ \{(f_u = 1), (f_v = 1)\} \mid uv \notin E(G) \}$ and let $S_{(1)} = (T_{(1)}, 1)$. For every $1 \leq i \leq k$, let $S^i_{(2)} = (\{ \{(f_v = 1)\} \mid v \in V_i \}, 0)$ and let $S^i_{\perp} = (\emptyset, 0)$.

Clearly, the reduction can be achieved in polynomial-time. It remains to show that G has a k-clique if and only if $\mathcal{S}$ classifies at least one example positively, that sets at most k features to 1. Towards showing the forward direction, let $C = \{ v_1, \ldots, v_k \}$ be a k-clique of G. We claim that the example e that is 1 exactly at the features $\{ f_{v_1}, \ldots f_{v_k} \}$ (and otherwise 0) is classified positively by $\mathcal{S}$. By construction, e is positively classified by every DS $S^i_{(2)}$. Moreover, since C is a k-clique then no term from $T_{(1)}$ is satisfied by e, which means that e is classified positively by $S_{(1)}$. Therefore, e is classified positively by $\mathcal{S}$.

Towards showing the reverse direction, suppose that there is an example e, that sets at most k features to 1, which is classified positively by $\mathcal{S}$. Note that all $S^i_{\perp}$ classify all examples negatively. Therefore DSs $S_{(1)}, S^1_{(2)}, \ldots, S^k_{(2)}$ classify e positively. Since e is classified positively by $S^i_{(2)}$, we get e sets exactly one feature from $\{ f_v \mid v \in V_i \}$ to 1. Moreover, since e is classified positively by $S_{(1)}$, all the vertices $v \in V(G)$ with $e(f_v) = 1$ form a clique in G. Therefore, $C = \{v | e(f_v) = 1 \wedge v \in V(G)\}$ is a k-clique of G.

Note that $\mathcal{S}$ classifies an example e_0 negatively. This implies that DS$_{\text{MAJ}}$-P-HOM(*ens_size*) is W[1]-hard even if *term_size* is constant. Moreover, by combining with Lemma 29 we get that DS$_{\text{MAJ}}$-LCXP$_{||}$(*xp_size* + *ens_size*) is W[1]-hard. □

Theorem 40. *Let* $\mathcal{M} \in \{\text{DS}_{\text{MAJ}}, \text{DL}_{\text{MAJ}}\}$ *and let* $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. $\mathcal{M}$-$\mathcal{P}_{\subseteq}$ *is* co-NP*-hard even if terms_elem + term_size is constant;* $\mathcal{M}$-LCXP$_{\subseteq}$ *is* NP*-hard even if terms_elem + term_size is constant;* $\mathcal{M}$-$\mathcal{P}_{||}$ *is* co-NP*-hard even if terms_elem + term_size + xp_size is constant;* $\mathcal{M}$-LCXP$_{||}$(*xp_size*) *is* W[1]*-hard even if terms_elem + term_size is constant.*

Proof. Again it suffices to show the theorem for the case that $\mathcal{M} = \text{DS}_{\text{MAJ}}$. We will use Lemma 30 to show the theorem by giving DSs M^0, M^1_f and $M^2_{\{f_1, f_2\}}$ satisfying the conditions of Lemma 30 as follows. We let M^0 be the DS $(\emptyset, 0)$. We let M^1_f be the DS given by $(\{\{(f = 1)\}\}, 0)$ and we let $M^2_{\{f_1,f_2\}}$ be the DS given by $(\{\{(f_1 = 0), (f_2 = 0)\}\}, 1)$. Note that M^0, M^1_f and $M^2_{\{f_1,f_2\}}$ have *term_size* at most 2 and *terms_elem* at most 1. All statements in the theorem now follow from Lemmas 28 to 30. □

6.3 OBDDs and their Ensembles

We are now ready to provide our hardness results for OBDDs and their ensembles $\text{OBDD}^{<}_{\text{MAJ}}$s and OBDD_{MAJ}s. While the proofs for OBDDs and $\text{OBDD}^{<}_{\text{MAJ}}$s follow along very similar lines as the corresponding proofs for DTs, the main novelty and challenge of this subsection are the much stronger hardness results for OBDD_{MAJ}s. Informally, we show that the satisfiability of any CNF formula ϕ can be modelled in terms of an ensemble of two OBDDs O_1 and O_2 each using a different ordering of the variables. In particular, it holds that both OBDDs classify an example positively if and only if the corresponding assignment satisfies ϕ. The main idea behind the construction of O_1 and O_2 is to make copies for every occurrence of a variable in ϕ and then use O_1 to verify that the assignment satisfies ϕ and O_2 to verify that all copies of every variable are assigned to the same value.

Theorem 41. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. OBDD_{MAJ}-$\mathcal{P}_{\subseteq}$ *is* co-NP*-hard even if ens_size + width_elem is constant;* OBDD_{MAJ}-$\text{LCXP}_{\subseteq}$ *is* NP*-hard even if ens_size + width_elem is constant;* OBDD_{MAJ}-$\mathcal{P}_{||}$ *is* co-NP*-hard even if ens_size + width_elem + xp_size is constant;* OBDD_{MAJ}-$\text{LCXP}_{||}$(*ens_size*) *is* W[1]*-hard even if ens_size + width_elem is constant.*

Proof. We provide a parameterized reduction from MCC to OBDD_{MAJ}-P-HOM, which is also a polynomial-time reduction. Given an instance (G, k) of MCC with a k-partition $V_1, \ldots, V_k$, we will construct an OBDD_{MAJ} $\mathcal{O}$ such that $|\mathcal{O}| = 3$, the maximal width of any OBDD in $\mathcal{O}$ at most 4 and G has a k-clique if and only if $\mathcal{O}$ classifies at least one example positively. Let $\{v_1, \ldots, v_n\} = V(G) = \bigcup_i V_i$. $\mathcal{O}$ will use $k + 2$ binary vertex features $f_a, f'_a, f^1_a, \ldots f^k_a$ for every vertex $v_a \in V(G)$ and three binary edge features $f'_{a,b}$, $f_{a,b}$ and $f_{b,a}$ for each edge $v_a v_b \in E(G) \wedge a < b$.

Before constructing $\mathcal{O}$, we start by providing some helpful auxiliary OBDDs that will correspond to operations on some subset F of features. For convenience, we define the fresh auxiliary feature $f_\#$ that will not be part of the constructed OBDD but instead only helps with its construction. We also fix an arbitrary ordering $<_F$ of the features in F. Let $next()$ be the successor function of the features in F w.r.t. $<_F$, i.e., for every $f \in F$, $next(f) = f'$ if f has a successor f' and $next(f) = f_\#$ otherwise. Let D_F be the directed acyclic graph with vertices $V(D_F) = \{ g^f_0, g^f_1, g^f_2 \mid f \in F \cup \{f_\#\} \}$ and source vertex $s = g_0^{\min_{<_F} F}$. Let ρ_F be the function giving by setting $\rho_F(g^f) = f$ for every $g^f \in V(D_F)$. We now define different OBDDs based on (D_F, ρ_F), which only differ in the edges and the assignment of 0-neighbors and 1-neighbors.

We first define the OBDD $O^!_F$ that classifies an example e positively if and only if exactly one of the features $f \in F$ appears positively in e. $O^!_F = (D^!_F, \rho_F)$ is obtained from (D_F, ρ_F) as follows. For each $f \in F$ and $i \in \{0, 1, 2\}$, let $g_i^{next(f)}$ and $g^{next(f)}_{\max\{i+1,2\}}$ be the 0-neighbor and 1-neighbor of g^f_i, respectively. We set t_1 to be $g_1^{f_\#}$ and we set t_0 to be equal to the vertex obtained after identifying $g_0^{f_\#}$ with $g_2^{f_\#}$. Note that the OBDD $O^!_F$ simulates a simple counter, i.e., if an example e reaches a node g^f_i it means that for i equal to 0, 1 and 2 there are no, exactly one and more than one, respectively, features in $\{f' | f' \in F \wedge f' <_F f\}$, for which e is 1.

We now define the OBDD $O^\exists_F = (D^\exists_F, \rho_F)$ that classifies an examples e positively if and only if there is a feature $f \in F$, where e is set to 1, i.e. $\exists_{f \in F} e(f) = 1$. The construction of $D^\exists_F$ is similar to $D^!_F$, but we set t_0 to be $g_0^{f_\#}$ and we set t_1 to be equal to the vertex obtained after identifying $g_1^{f_\#}$ with $g_2^{f_\#}$. This modification ensures that only examples with one or more features in F set to 1 reach t_1.

Next, we define the OBDD $O^{\Leftrightarrow\exists}_{\{f\}\cup F} = (D^{\Leftrightarrow\exists}_{\{f\}\cup F}, \rho^{\Leftrightarrow\exists}_{\{f\}\cup F})$ that classifies an examples e positively if and only if the special feature f is set to 1 by e if and only if there is a feature $f' \in F$ that appears positively in e, i.e., $e(f) = 1 \Leftrightarrow \exists_{f' \in F} e(f') = 1$. To obtain $D^{\Leftrightarrow\exists}_{\{f\}\cup F}$ we modify $D^\exists_F$ by replacing t_0 and t_1 with new vertices g^f_0 and g^f_1, respectively. Moreover for each $i \in \{0, 1\}$, let t_1 be i-neighbor of g^f_i and let t_0 be the other neighbor. The modification compares the result from $D^\exists_F$ with f.

We now define the OBDD $O^=_F = (D^=_F, \rho_{D^=_F})$ that classifies an example e positively if and only if all or none of the features $f \in F$ appears positively in e, i.e., $\forall_{f \in F} e(f) = 0$ or $\forall_{f \in F} e(f) = 1$. For each $f \in F$, let $g_2^{next(f)}$ be the 0-neighbor and 1-neighbor of g^f_2. Let f_{first} be the first feature in F. Let $g_0^{next(f_{first})}$ and $g_1^{next(f_{first})}$ be the 0-neighbor and 1-neighbor of the source vertex $s = g_0^{f_{first}}$, respectively. For each $f \in F \setminus \{f_{first}\}$ and $i \in \{0, 1\}$, let $g_i^{next(f)}$ be i-neighbor of g^f_i and let $g_2^{next(f)}$ be the other neighbor. We set t_0 to be $g_2^{f_\#}$ and we set t_1 to be equal to the vertex obtained after identifying $g_0^{f_\#}$ with $g_1^{f_\#}$. Note that vertices in $D^=_F$ have different roles compared to other reductions, which are to store information about $e(f_{first})$ and the negative result. If example e reaches node g^f_i and $i \in \{0, 1\}$, it implies that for every $f' \in F \wedge f' < f$, $e(f_{first}) = e(f') = i$. If example e reaches node g^f_2, it implies that there exists $f' \in F \wedge f' <_F f$, such that $e(f_{first}) \neq e(f')$.

Note that all of the above OBDDs have width equal to 3. Before showing our main reduction we need to introduce a tool to help us merge the result of several OBDDs into one.

Let $O_1, \cdots, O_m$ be OBDDs such that no two of them share a feature. We will construct the OBDD $O = (D, \rho)$ that classifies an examples e positively if and only if for every $1 \leq i \leq m$, O_i classifies e positively. For each $1 \leq i \leq m$, let $O_i = (D_i, \rho_i)$, F_i be set of features of O_i with an order $<_{F_i}$ and let s^i, t^i_0, t^i_1 be the special nodes s, t_0, t_1 of D_i, respectively. Let $F = \bigcup F_i$ and let $<_F$ be the ordering of the features in F such that for every $f_1 \in F_i$ and $f_2 \in F_j$, if $f_1 <_F f_2$ then $i < j$ or $i = j$ and $f_1 <_{F_i} f_2$.

Let D be the graph obtained from the union of the graphs D_i after adding the new vertices $\{t_0, t_1\} \cup \{g^f_\perp | f \in F \cup \{f_\#\}\}$ as follows. For every $f \in F$, let $g_\perp^{next(f)}$ be the

0-neighbor and 1-neighbor of $g^f_\bot$. For every $1 \leq i < m$, we identify s^{i+1} with t^i_1 and $g^{f^{i+1}_{first}}_\bot$ with t^i_0, where f^{i+1}_{first} is the $<_{F_{i+1}}$-smallest feature in F_{i+1}. Moreover, we set t_1 to be t^m_1 and we set t_0 to be equal to the vertex obtained after identifying $g^{f_\#}_\bot$ with t^m_0. Note that if example e reaches a node $g^f_\bot$ and $f \in F_i$, it means there exists $1 \leq j < i$ such that O_j classifies e negatively. Moreover, the width of O is equal maximal width of O_i plus one.

Finally, we are ready to provide our reduction from MCC. We will construct $\text{OBDD}^<_{\text{MAJ}}$ $\mathcal{O} = \{O_1, O_2\, O_3\}$, which will classify an example e positively if and only if all of the following items are true:

(1) for every $v_a \in V(G)$, all or none of the vertex features $f_a, f'_a, f^1_a, \dots f^k_a$ are set to 1 in e,
(2) for every $v_a v_b \in E(G)$ and $a < b$, all or none of the edge features $f'_{a,b}$, $f_{a,b}$, $f_{b,a}$ are set to 1 in e,
(3) for every $1 \leq i \leq k$, exactly one feature from $F^i = \{f'_a | v_a \in V_i\}$ is set to 1 in e.
(4) for every $1 \leq i < j \leq k$, exactly one feature from $F^{i,j} = \{f'_{\min\{a,b\},\max\{a,b\}} | v_a \in V_i \wedge v_b \in V_j \wedge v_a v_b \in E(G)\}$ is set to 1 in e,
(5) for every $1 \leq i, j \leq k \wedge i \neq j$ and $v_a \in V_i$, $e(f^j_a) = 1$ if and only if at least one feature from $F^j_a = \{f_{a,b} | v_b \in V_j \wedge v_a v_b \in E(G)\}$ is set to 1 in e

For every item (i) from the list above we create $O_{(i)}$ which classify an example e positively if and only if it satisfies property (i).

$$O_{(1)} = \bigwedge_{v_a \in V(G)} O^{=}_{\{f_a, f'_a, f^1_a, \dots f^k_a\}}$$

$$O_{(2)} = \bigwedge_{v_a v_b \in E(G) \wedge a < b} O^{=}_{\{f'_{a,b}, f_{a,b}, f_{b,a}\}}$$

$$O_{(3)} = \bigwedge_{1 \leq i \leq k} O^{!}_{F^i}$$

$$O_{(4)} = \bigwedge_{1 \leq i < j \leq k} O^{!}_{F^{i,j}}$$

$$O_{(5)} = \bigwedge_{1 \leq i,j \leq k \wedge i \neq j \wedge v_a \in V_i} O^{\Leftrightarrow \exists}_{\{f^j_a \cup F^j_a\}}$$

Let $O_1 = O_{(1)} \wedge O_{(2)}$, $O_2 = O_{(3)} \wedge O_{(4)} \wedge O_{(5)}$ and let O_3 be a trivial OBDD that classifies every example negatively.

Note that $\mathcal{O}$ classifies an example e positively if and only if O_1 and O_2 classifies e positively, i.e., if and only if for every $1 \leq i \leq 5$, $O_{(i)}$ classifies e positively. Moreover, the width of O_1 and O_2 is 4, because they are made from the logical conjunction of OBDDs $O^{!}_F$, $O^{\exists}_F$, $O^{\Leftrightarrow \exists}_{\{f\} \cup F}$ and $O^{=}_F$ of width 3. This completes the description of our reduction, which can clearly be achieved in polynomial-time. It remains to show that G has a k-clique if and only if $\mathcal{O}$ classifies at least one example positively.

Towards showing the forward direction, let $C = \{v_{a_1}, \dots, v_{a_k}\}$ be a k-clique of G, where $v_{a_i} \in V_i$ for every i with $1 \leq i \leq k$. We claim that the example e that is 1 exactly at the features $\{f_{a_i}, f'_{a_i}, f^j_{a_i} \mid 1 \leq i, j \leq k\} \cup \{f'_{a_i,a_j}, f_{a_i,a_j}, f_{a_j,a_i} \mid 1 \leq i, j \leq k \wedge i < j\}$ (and otherwise 0) is classified positively by $\mathcal{O}$. By construction, e is positively classified by $O_{(1)}$ and $O_{(2)}$, since C is a k-clique also $O_{(3)}$, $O_{(4)}$ and $O_{(5)}$. Therefore, e is classified positively by O_1 and O_2, which represents the majority of the DTs in $\mathcal{O}$.

Towards showing the reverse direction, suppose that there is an example e that is classified positively by $\mathcal{O}$. Because e has to be classified positively by the majority of OBDD_{MAJ} in $\mathcal{O}$, we obtain that e has to be classified positively by O_1 and O_2. Since e is classified positively by O_1, we obtain all of the features that correspond to a specific vertex or a specific edge are assigned the same way. Moreover, since $O_{(5)}$ and O_1 classify e positively, this means that if edge feature $f_{a,b}$ is set to 1, then both vertex features f_a and f_b are set to 1. Since e is classified positively by O_1, $O_{(3)}$ and $O_{(4)}$, it means that the edge features of exactly $\binom{k}{2}$ edges are set to 1 and that the vertex features of exactly k vertices are set to 1. Therefore, $C = \{v_a | e(f_a) = 1 \wedge v_a \in V(G)\}$ is a k-clique of G.

Note that the all-zero example is classified negatively by $\mathcal{O}$ and if there is an example e such that $\mathcal{O}$ classify e positively then e contains exactly $3\binom{k}{2} + k(k+2)$ positively assigned features. It implies that OBDD_{MAJ}-P-HOM is W[1]-hard even if *ens_size* equals 3 and *width_elem* equals 4, which combined with Lemma 29 shows that OBDD_{MAJ}-$\text{LCX}_{\text{P}_{||}}$ is W[1]-hard parameterized by *xp_size* even if *ens_size* + *width_elem* is constant. Moreover, OBDD_{MAJ}-HOM is NP-hard even if *ens_size* equals 3 and *width_elem* equals 4, which combined with Lemma 28 shows every other statement in the theorem. □

Lemma 42. *Let $\mathcal{T}$ be an ordered* DT *for the ordering $<_{\mathcal{T}}$. There is an* $\text{OBDD}^{<_{\mathcal{T}}}$ $O^{\mathcal{T}}$ *that can be constructed in polynomial-time such that $\mathcal{T}(e) = O^{\mathcal{T}}(e)$ for every example e.*

Proof. Let $\mathcal{T} = (T, \lambda)$. We define $O^{\mathcal{T}} = (D^{\mathcal{T}}, \rho^{\mathcal{T}})$ as follows. Let $D^{\mathcal{T}}$ be the directed acyclic graph obtained from T after directing all edges away from the root and identifying every b-leaf with a new vertex t_b for every $b \in \{0, 1\}$. Moreover, let $\rho^{\mathcal{T}}(v) = \lambda(v)$ for every inner vertex of $D^{\mathcal{T}}$. This completes the construction of our OBDD, which because of (Mengel and Slivovsky 2021, Observation 1) can also be turned into an equivalent complete OBDD. □

A special case of the following theorem, the NP-hardness of $\text{LAX}_{\text{P}_{||}}$ for FBDDs, i.e., "free" BDDs, was shown by Barceló et al. (2020).

Theorem 43. *Let $\mathcal{P} \in \{\text{LAX}_{\text{P}}, \text{GAX}_{\text{P}}, \text{GCX}_{\text{P}}\}$.* OBDD-$\mathcal{P}_{||}$ *is* NP*-hard and* OBDD-$\mathcal{P}_{||}$(*xp_size*) *is* W[2]*-hard.*

Proof. Combining Theorem 32 with Lemma 42 shows that OBDD-$\text{LAX}_{\text{P}_{||}}$ is W[2]-hard parameterized by *xp_size*.

Next, we will provide a polynomial-time reduction from OBDD-$\text{LAX}_{\text{P}_{||}}$ to OBDD-$\text{GAX}_{\text{P}_{||}}$ and one from OBDD-$\text{GAX}_{\text{P}_{||}}$ to OBDD-$\text{GCX}_{\text{P}_{||}}$. Let (O, e, k) be an instance for OBDD-$\text{LAX}_{\text{P}_{||}}$, let $F = F(O)$ and let $<_F$ be the order

of the features as they occur on paths of O. We now construct the OBDD$^{<_F}$ $O^k_e = (D^k_e, \rho^k_e)$ (which uses the order $<_F$) that classifies an example e' as $O(e)$ if and only if there are at least k features that e and e' agree on, i.e., $k \leq |\{f : e(f) = e'(f) \wedge f \in F\}|$. For convenience, we will again use the auxiliary feature $f_\#$, which will not occur in the final OBDD. Let $next()$ be the successor function of the features in F w.r.t. $<_F$, i.e., for every $f \in F$, $next(f) = f'$ if f has a successor f' and $next(f) = f_\#$ otherwise. Let D^k_e be the directed acyclic graph with vertices $V(D^k_e) = \{\, g^f_i \mid 0 \leq i \leq k \wedge f \in F \cup \{f_\#\}\,\}$. Let ρ^k_e be the function defined by setting $\rho^k_e(g^f) = f$ for every $g^f \in V(D^k_e)$. For each $f \in F$, let $g^{next(f)}_k$ be the 0-neighbor and 1-neighbor of g^f_k. For each $f \in F$ and $0 \leq i < k$, let $g^{next(f)}_{i+1}$ be the $e(f)$-neighbor of g^f_i and let $g^{next(f)}_i$ be the other neighbor. We set t_1 to be $g^{f_\#}_k$ and we set t_0 to be equal to the vertex obtained after identifying all vertices in $\{g^{f_\#}_i | 0 \leq i < k\}$. Note that the graph D^k_e simulates a simple counter, i.e., if an example e' reaches node g^f_i and $i < k$ it means that there are exactly i features from $\{f' | f' \in F \wedge f' <_F f\}$ that e and e' agree on.

Let $\mathcal{O}^{<_F}$ be the OBDD$^{<_F}_{\text{MAJ}}$ given by $\mathcal{O}^{<_F} = \{O, O^k_e, O_\perp\}$, where $O_\perp$ is a trivial OBDD$^{<_F}$ that classifies every example $1 - O(e)$. We apply Lemma 26 to $\mathcal{O}^{<_F}$ to create an OBDD$^{<_F}$ O' that is equivalent with $\mathcal{O}^{<_F}$. Note this reduction can be achieved in polynomial-time.

We are now ready to prove that the following problems are equivalent:

(1) (O, e, k) is a yes-instance of OBDD-LAXP$_{||}$.

(2) $(O', O(e), k)$ is a yes-instance of OBDD-GAXP$_{||}$.

(3) $(O', 1 - O(e), k)$ is a yes-instance of OBDD-GCXP$_{||}$.

It is obvious that (2) and (3) are equivalent.

Towards showing that (1) $\implies$ (2), let $A \subseteq F$ be a local abductive explanation for e w.r.t. O of size at most k, i.e., every example e' that agrees with e on the features in A is classified as $O(e)$ by O. Let A' be a superset of A that contains exactly k features and let $\tau : A' \to \{0, 1\}$ be the assignment defined by setting $\tau(f) = e(f)$ for every $f \in A'$. We claim that τ is a global abductive explanation for $O(e)$ w.r.t. O' of size at most k, i.e., every example e' that agrees with τ is classified as $O(e)$ by O'. But this clearly holds because both O and O^k_e classify e as $O(e)$.

Towards showing that (2) $\implies$ (1), let $\tau : A \to \{0, 1\}$ for some $A \subseteq F$ with $|A| \leq k$ be a global abductive explanation for $O(e)$ w.r.t. O', i.e., every example e' that agrees with τ is classified as $O(e)$ by O' and therefore also by O and O^k_e. But then, τ has to agree with e on at least k features, because otherwise there would be an example that agrees with τ but is not classified as $O(e)$ by O^k_e. But then, A is a local abductive explanation for e w.r.t. O. □

Theorem 44. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. OBDD$^<_{\text{MAJ}}$-$\mathcal{P}_\subseteq$ *is* co-NP*-hard even if width_elem + size_elem is constant;* OBDD$^<_{\text{MAJ}}$-LCXP$_\subseteq$ *is* NP*-hard even if width_elem + size_elem is constant;* OBDD$^<_{\text{MAJ}}$-$\mathcal{P}_{||}$ *is* co-NP*-hard even if width_elem + size_elem + xp_size is constant;* OBDD$^<_{\text{MAJ}}$-LCXP$_{||}$(*xp_size*) *is* W[1]*-hard even if width_elem + size_elem is constant;*

Proof. Let F be a set of features and let $<_F$ be an arbitrary ordering of the features in F.

We will use Lemma 30 to show the theorem by giving OBDD$^{<_F}$s M^0, M^1_f and $M^2_{\{f_1,f_2\}}$ satisfying the conditions of Lemma 30 as follows. We let M^0 be the trivial OBDD$^{<_F}$ that classifies every example negatively. We let M^1_f be the OBDD$^{<_F}$ (D^1_f, ρ^1_f), such that $V(D^1_f) = \{g, t_0, t_1\}$, $\rho^1_f(g) = f$, g is the source vertex, and t_0 and t_1 is the 0-neighbor and 1-neighbor of g, respectively. We let $M^2_{f_1,f_2}$ be the OBDD$^{<_F}$ $(D^2_{f_1,f_2}, \rho^2_{f_1,f_2})$ such that $V(D^2_{f_1,f_2}) = \{g^{f_1}, g^{f_2}, t_0, t_1\}$, $\rho^2_{f_1,f_2}(g^{f_1}) = f_1$ and $\rho^2_{f_1,f_2}(g^{f_2}) = f_2$. Here, we assume that w.l.o.g. $f_1 <_F f_2$. We let g^{f_1} be the source vertex. We let t_1 and g^{f_2} be the 0-neighbor and 1-neighbor of g^{f_1}, respectively. Moreover, we let t_1 and t_0 be the 0-neighbor and 1-neighbor of g^{f_2}, respectively.

Note that M^0, M^1_f and $M^2_{\{f_1,f_2\}}$ have *size_elem* at most 4 and *width_elem* equal 2. All statements in the theorem now follow from Lemmas 28 to 30. □

Theorem 45. *Let* $\mathcal{P} \in \{\text{LAXP}, \text{GAXP}, \text{GCXP}\}$. OBDD$^<_{\text{MAJ}}$-$\mathcal{P}_\subseteq$(*ens_size*) *is* co-W[1]*-hard;* OBDD$^<_{\text{MAJ}}$-LCXP$_\subseteq$(*ens_size*) *is* W[1]*-hard;* OBDD$^<_{\text{MAJ}}$-$\mathcal{P}_{||}$(*ens_size*) *is* co-W[1]*-hard even if xp_size is constant;* OBDD$^<_{\text{MAJ}}$-LCXP$_{||}$(*ens_size* + *xp_size*) *is* W[1]*-hard.*

Proof. All statements in the theorem follow from Lemmas 28, 29, 34, 42. □

7 Conclusion

We have developed an in-depth exploration of the parameterized complexity of explanation problems in various machine learning (ML) models, focusing on models with transparent internal mechanisms. By analyzing different models and their ensembles, we have provided a comprehensive overview of the complexity of finding explanations in these systems. These insights are crucial for understanding the inherent complexity of different ML models and their implications for explainability.

Among our findings, some results stand out as particularly unexpected. For instance, while DT$_{\text{MAJ}}$ and OBDD$^<_{\text{MAJ}}$s are seemingly different model types, our results show that they behave similarly w.r.t. tractability for explanation problems. On the other hand, it seems surprising that many of the tractability results that hold for DTs and OBDDs do not carry over to seemingly simpler models such as DSs and DLs. For instance, while all variants of LCXP are polynomial-time for DTs and OBDDs, this is not the case for DSs or DLs. Nevertheless, we obtain interesting FPT-algorithms for DL-LCXP$_{||}$ (Theorem 18). OBDD$_{\text{MAJ}}$ stands out as the hardest model for computing explanations by far, which holds even for models with only two ensemble elements. From a complexity point of view, DT-GAXP$_{||}$ provides the rare scenario where a problem is known as W[1]-hard but not confirmed to be NP-hard (Theorem 33).

Looking ahead, there are several promising directions for future research. First, we aim to extend our complexity classification to Sequential Decision Diagrams (Darwiche 2011) or even FBDDs, which offer a more succinct representation than OBDDs (Bova 2016). This extension could provide further insights into the complexity of explanations in more compact ML models. Secondly, we propose to explore other problem variations, such as counting different types of explanations or finding explanations that meet specific constraints beyond just the minimum ones (Barceló et al. 2020). Lastly, the concept of weighted ensembles presents an intriguing avenue for research. While the hardness results we established likely still apply, the tractability in the context of weighted ensembles needs to be clarified and warrants further investigation. It would be interesting to see how our results hold up when considering polynomial-sized weights.

In summary, our work marks a significant stride in the theoretical understanding of explainability in AI. This research responds to the practical and regulatory demand for transparent, interpretable, and trustworthy AI systems by offering a detailed complexity analysis across various ML models. As the field of XAI evolves, our study lays a foundational groundwork for future research and the development of more efficient explanation methods in AI.

Acknowledgements

Stefan Szeider acknowledges support by the Austrian Science Fund (FWF) within the projects 10.55776/P36688, 10.55776/P36420, and 10.55776/COE12. Sebastian Ordyniak was supported by the Engineering and Physical Sciences Research Council (EPSRC) (Project EP/V00252X/1).

References

Barceló, P.; Monet, M.; Pérez, J.; and Subercaseaux, B. 2020. Model interpretability through the lens of computational complexity. *Proc. NeurIPS 2020* 33:15487–15498.

Bergougnoux, B.; Dreier, J.; and Jaffke, L. 2023. A logic-based algorithmic meta-theorem for mim-width. *Proc. SODA 2023* 3282–3304.

Bova, S. 2016. SDDs are exponentially more succinct than OBDDs. *Proc. AAAI 2016* 929–935.

Carvalho, D. V.; Pereira, E. M.; and Cardoso, J. S. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8(8).

Chan, H., and Darwiche, A. 2003. Reasoning about bayesian network classifiers. *Proc. UAI 2003* 107–115.

Commission, E. 2019. *Ethics guidelines for trustworthy AI*. Publications Office, European Commission and Directorate-General for Communications Networks, Content and Technology.

Commission, E. 2020. *White Paper on Artificial Intelligence: a European approach to excellence and trust*. Publications Office, European Commission and Directorate-General for Communications Networks, Content and Technology.

Corneil, D. G., and Rotics, U. 2001. On the relationship between clique-width and treewidth. In Brandstädt, A., and Le, V. B., eds., *Graph-Theoretic Concepts in Computer Science, 27th International Workshop, WG 2001, Boltenhagen, Germany, June 14-16, 2001, Proceedings*, volume 2204 of *Lecture Notes in Computer Science*, 78–90. Springer.

Darwiche, A., and Ji, C. 2022. On the computation of necessary and sufficient explanations. *Proc. AAAI 2022* 5582–5591.

Darwiche, A. 2011. SDD: A new canonical representation of propositional knowledge bases. *Proc. IJCAI 2011* 819–826.

Diestel, R. 2000. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. New York: Springer Verlag, 2nd edition.

Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of parameterized complexity*. Texts in Computer Science. Springer Verlag.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2019. A survey of methods for explaining black box models. *ACM Computing Surveys* 51(5):93:1–93:42.

Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; and Samek, W. 2020. Explainable AI methods - A brief overview. *Proc. xxAI@ICML 2020* 13200:13–38.

Ignatiev, A.; Narodytska, N.; NicholasAsher; and Marques-Silva, J. 2020. From contrastive to abductive explanations and back again. *Proc. AIxIA 2020* 12414:335–355.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019. Abduction-based explanations for machine learning models. *Proc. AAAI 2019* 1511–1519.

Jha, A. K., and Suciu, D. 2012. On the tractability of query compilation and bounded treewidth. *Proc. ICDT 2012* 249–261.

Lipton, Z. C. 2018. The mythos of model interpretability. *Communications of the ACM* 61(10):36–43.

Lisboa, P. J. G.; Saralajew, S.; Vellido, A.; Fernández-Domenech, R.; and Villmann, T. 2023. The coming of age of interpretable and explainable machine learning models. *Neurocomputing* 535:25–39.

Marques-Silva, J. 2023. Logic-based explainability in machine learning. *Reasoning Web. Causality, Explanations and Declarative Knowledge* 24–104.

Mengel, S., and Slivovsky, F. 2021. Proof complexity of symbolic QBF reasoning. *Proc. SAT* 12831:399–416.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267:1–38.

Molnar, C. 2023. *Interpretable Machine Learning*. Lulu.com.

OECD. 2023. The state of implementation of the OECD AI principles four years on. Technical Report 3, The Organisation for Economic Co-operation and Development.

Ordyniak, S.; Paesani, G.; Rychlicki, M.; and Szeider, S. 2024. Explaining decisions in ML models: a parameterized complexity analysis. *arXiv* (2407.15780).

Ordyniak, S.; Paesani, G.; and Szeider, S. 2023. The parameterized complexity of finding concise local explanations. *Proc. IJCAI 2023* 3312–3320.

Oum, S., and Seymour, P. D. 2006. Approximating clique-width and branch-width. *Journal of Combinatorial Theory* 96(4):514–528.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "why should I trust you?": Explaining the predictions of any classifier. *Proc. KDD 2016* 1135–1144.

Shih, A.; Choi, A.; and Darwiche, A. 2018. A symbolic approach to explaining Bayesian network classifiers. *Proc. IJCAI 2018* 5103–5111.