Even-if Explanations: Formal Foundations, Priorities and Complexity

Gianvincenzo Alfano, Sergio Greco, Domenico Mandaglio, Francesco Parisi, Reza Shahbazian, Irina Trubitsyna

Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, Italy
{g.alfano, greco, d.mandaglio, fparisi, i.trubitsyna}@dimes.unical.it reza.shahbazian@unical.it

Abstract

Explainable AI has received significant attention in recent years. Machine learning models often operate as black boxes, lacking explainability and transparency while supporting decision-making processes. Local post-hoc explainability queries attempt to answer why individual inputs are classified in a certain way by a given model. While there has been important work on counterfactual explanations, less attention has been devoted to semifactual ones. In this paper, we focus on local post-hoc explainability queries within the semifactual 'even-if' thinking and their computational complexity among different classes of models, and show that both linear and tree-based models are strictly more interpretable than neural networks. After this, we introduce a preference-based framework enabling users to personalize explanations based on their preferences, both in the case of semifactuals and counterfactuals, enhancing interpretability and user-centricity. Finally, we explore the complexity of several interpretability problems in the proposed preference-based framework and provide algorithms for polynomial cases.

Introduction

The extensive study of counterfactual 'if only' thinking, exploring how things might have been different, has been a focal point for social and cognitive psychologists (Kahneman and Tversky 1981; McCloy and Byrne 2002). Consider a negative event, such as taking a taxi and due to traffic arriving late to a party. By analyzing this situation, an individual (e.g. Alice) might engage in counterfactual thinking by imagining how things could have unfolded differently, such as, 'if only Alice had not taken the taxi, she would not have arrived late at the party'. This type of counterfactual thinking, where an alternative scenario is imagined, is a common aspect of daily life. In such a case the counterfactual scenario negates both the event's cause (antecedent) and its outcome, presenting a false cause and a false outcome that are temporarily considered as true (e.g., Alice took the taxi and arrived late).

Counterfactual thinking forms the basis for crafting counterfactual explanations, which are crucial in automated decision-making processes. These explanations leverage imagined alternative scenarios, aiding users in understanding why certain outcomes occurred and how different situations might have influenced decisions. Counterfactual explanations empower users to grasp the rationale behind decisions, fostering transparency and user trust in these systems. Several definitions of counterfactual explanations exist in the literature (Guidotti 2022; Jiang et al. 2024; Wu, Wu, and Barrett 2023). According to most of the literature, counterfactuals are defined as the minimum changes to apply to a given instance to let the prediction of the model be different (Barceló et al. 2020).

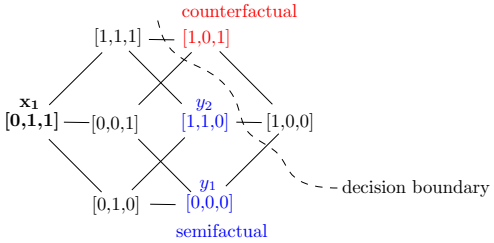

Figure 1: Binary classification model $\mathcal{M}$: $step(\mathbf{x} \cdot [-2, 2, 0] + 1)$ of Example 1 (hiring scenario). The binary feature f_1 (resp., f_2 and f_3) represents part-time employment contract (resp., salary lower than 5K$, and on site-working).

While significant attention in AI has been given to counterfactual explanations, there has been a limited focus on the equally important and related semifactual 'even if' explanations (Aryal and Keane 2023; Kenny and Huang 2023), though they have been investigated much more in cognitive sciences. While counterfactuals explain what changes to the input features of an AI system change the output decision, semifactuals show which input feature changes do not change a decision outcome. Considering the above-mentioned situation where Alice took the taxi and arrived late at the party, we might analyze 'even if' scenarios envisioning how things could have remained the same, such as, "even if Alice had not taken a taxi, she would have still arrived late at the party".

Sharing the same underlying idea of counterfactuals, we define semifactuals as the maximum changes to be applied to a given instance while keeping the same prediction. Indeed, the larger the feature differences asserted in the semifactual, the better (more convincing) the explanation (Aryal and Keane 2023). Semifactual explanations incorporating more feature changes offer several benefits, including improved decision-

making and enhanced interpretability. For decision-makers, understanding the extent of changes that do not affect the outcome can aid in optimizing processes. For instance, in resource allocation, knowing the maximum allowable changes helps in making adjustments without compromising results. Such explanations provide a comprehensive understanding of a model's decision boundary by revealing how the model processes information and identifying which input aspects are critical for maintaining the decision. Semifactuals indicate which feature-sets are not relevant for classification, as they can be changed without altering the outcome. Also, considering a large number of feature changes in the semifactual intuitively captures the desire of an agent to have more flexibility and favorable conditions (represented by features changed), while keeping the (positive) status assigned to it by the model. Consider the following hiring scenario.

Example 1. Consider the binary and linear classification model $\mathcal{M} : \{0,1\}^3 \rightarrow \{0,1\}$ shown in Figure 1, where $\mathcal{M}$ is defined as $step(\mathbf{x} \cdot [-2,2,0] + 1)$ and the input $\mathbf{x} = [x_1, x_2, x_3]$ denotes an applicant (also called user) defined by means of the following three features:

- $f_1 =$ "part-time job";
- $f_2 =$ "requested (monthly) salary $<$ 5K$";
- $f_3 =$ "on-site job".

For any instance $\mathbf{x} \in \{0,1\}^3$ we have that $\mathcal{M}(\mathbf{x}) = 0$ if $\mathbf{x} = [1,0,1]$ or $\mathbf{x} = [1,0,0]$, and $\mathcal{M}(\mathbf{x}) = 1$ otherwise. Intuitively, this means that the company's AI model does not approve the application only when the user applies for a part-time job and the requested salary is no less than 5K$.

Consider a user $\mathbf{x}_1$ that applies for a full-time and on-site job, and the requested salary is lower than 5K$ (i.e., $\mathbf{x}_1 = [0,1,1]$), we have that $\mathbf{y}_1 = [0,0,0]$ and $\mathbf{y}_2 = [1,1,0]$ are semifactual of $\mathbf{x}_1$ w.r.t. $\mathcal{M}$ at maximum distance (i.e., 2) from $\mathbf{x}_1$ in terms of number of features changed. Intuitively, $\mathbf{y}_1$ represents the fact that 'the user $\mathbf{x}_1$ will be hired *even if* (s)he had requested for a remote job and the requested salary was greater than or equal to 5K$', while $\mathbf{y}_2$ represents 'the user $\mathbf{x}_1$ will be hired *even if* (s)he had applied for a remote and part-time job'. □

We would point out that counterfactuals and semifactuals are strongly connected and they should be considered together in eXplainable AI (XAI) as they describe which changes to feature-inputs of a black-box AI system result in changes to or confirmation to a decision-outcome, that is both contribute in understanding the presence of a decision boundary in the classification process. Taking for instance our running example, whose feature-values are shown in Figure 1, where edges represent changes of a unique feature value, the decision boundary can be described by considering both counterfactuals and semifactuals.

As highlighted in the previous example, multiple semifactuals can exist for each given instance. In these situations, a user may prefer one semifactual to another, by expressing preferences over features so that the *best* semifactuals will be selected, as shown in the following example.

Example 2. Continuing with Example 1, suppose that the user $\mathbf{x}_1$ looks for another opportunity and prefers to change feature f_2 rather than f_1 (irrespective of any other change), that is (s)he prefers semifactuals with $f_2 = 0$ rather than those with $f_1 = 1$. Thus, (s)he would prefer to still get hired by changing the salary to be greater than or equal to 5K$ (obtaining $\mathbf{y}_1$); if this cannot be accomplished, then (s)he prefers to get it by changing the job to part-time (i.e. $\mathbf{y}_2$). □

Prioritized reasoning in AI, focusing on incorporating user preferences, represents a pivotal advancement in the field, enhancing adaptability and user-centricity of AI systems. Traditional AI models rely on predefined rules or optimization criteria to generate outcomes, often overlooking the nuanced nature of user-specific preferences (Rossi, Venable, and Walsh 2011; Santhanam, Basu, and Honavar 2016). Prioritized reasoning addresses this limitation by introducing a mechanism that allows users to express their preferences, thereby guiding AI systems to prefer specific factors over others in the decision-making processes.

One key aspect of prioritized reasoning is its applicability across diverse AI domains, spanning machine learning (Kapoor et al. 2012), natural language processing (Bakker et al. 2022), and recommendation systems (Zhu et al. 2022).

Our work contributes to prioritized reasoning within explainable AI in the presence of user's preference conditions related to features. These preferences are exploited to generate semifactual and counterfactual explanations that align most closely with the user-specified criteria. In particular, preferences are applied similarly to what has been proposed in the well-known Answer Set Optimization (ASO) approach (Brewka, Niemelä, and Truszczynski 2003).

Contributions Our main contributions are as follows.

- We formally introduce the concepts of semifactual over three classes of models: (i) *perceptrons* (ii) *free binary decision diagrams (FBBDs)*, and (iii) *multi-layer perceptrons (MLP)*, intuitively encoding local post-hoc explainable queries within the even-if thinking setting. Herein, the term 'local' refers to explaining the output of the system for a particular input, while 'post-hoc' refers to interpreting the system after it has been trained.
- We investigate the computational complexity of interpretability problems concerning semifactuals, showing that they are not more difficult than analogous problems related to counterfactuals (Barceló et al. 2020).
- We introduce a framework that empowers users to prioritize explanations according to their subjective preferences. That is, users can specify preferences for altering specific features over others within explanations. This approach enriches the explanation process, enabling users to influence the selection of the most favorable semifactuals (called "best" semifactuals), thereby augmenting the interpretability and user-centricity of the resulting outputs. Notably, the proposed framework also naturally encompasses preferences over counterfactuals.
- We investigate the complexity of several interpretability problems related to best semifactuals and best counterfactuals. Table 1 summarizes our complexity results. Finally, focusing on a restricted yet expressive class of feature preferences, we identify tractable cases for which we propose algorithms for their computation.

Full proofs can be found in (Alfano et al. 2024a).

Preliminaries

We start by recalling the key concepts underlying counterfactual and semifactual explanations, and then we recall the main complexity classes used in the paper.

Classification Models. A (binary classification) model is a function $\mathcal{M} : \{0,1\}^n \to \{0,1\}$, specifically focusing on instances whose features are represented by binary values. Constraining inputs and outputs to booleans simplifies our context while encompassing numerous relevant practical scenarios. A class of models is just a way of grouping models together. An instance $\mathbf{x}$ is a vector in $\{0,1\}^n$ and represents a possible input for a model. We recall 3 significant categories of ML models that will be the ones we will focus on.

A *Binary Decision Diagram* (BDD) $\mathcal{M} = (V, E, \lambda_V, \lambda_E)$ (Wegener 2004) is a rooted directed acyclic graph (V, E) where (i) leave nodes are either labeled 1 (also denoted as $\top$) or 0 (also denoted as $\bot$), (ii) internal nodes are labeled by function λ_V with a value from $\{1, \ldots, n\}$, and (iii) each internal node has two outgoing edges labeled by function λ_E as 1 and 0, respectively. Each instance $\mathbf{x} = [x_1, \ldots, x_n] \in \{0,1\}^n$ uniquely maps to a path $p_{\mathbf{x}}$ in $\mathcal{M}$. This path adheres to the following condition: for every non-leaf node u in $p_{\mathbf{x}}$ labeled i, the path goes through the edge labeled with x_i. $|\mathcal{M}|$ denotes the size of $\mathcal{M}$, representing the number of edges. A binary decision diagram $\mathcal{M}$ is *free* (FBDD) if for every path from the root to a leaf, no two nodes on that path have the same label. A *decision tree* is simply an FBDD whose underlying graph is a tree.

A *multilayer perceptron (MLP)* $\mathcal{M}$ with k layers is defined by a sequence of weight matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(k)}$, bias vectors $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(k)}$, and activation functions $a^{(1)}, \ldots, a^{(k)}$. Given an instance $\mathbf{x}$, we inductively define $\mathbf{h}^{(i)} = a^{(i)}(\mathbf{h}^{(i-1)}\mathbf{W}^{(i)} + \mathbf{b}^{(i)})$ with $i \in \{1, \ldots, k\}$, assuming that $\mathbf{h}^{(0)} = \mathbf{x}$. The output of $\mathcal{M}$ on $\mathbf{x}$ is defined as $\mathcal{M}(\mathbf{x}) = \mathbf{h}^{(k)}$. In this paper we assume all weights and biases to be rational numbers, i.e. belonging to $\mathbb{Q}$. We say that an MLP as defined above has $(k-1)$ *hidden layers*. The *size* of an MLP $\mathcal{M}$, denoted by $|\mathcal{M}|$, is the total size of its weights and biases, in which the size of a rational number $\frac{p}{q}$ is $log_2(p) + log_2(q)$ (with the convention that $log_2(0) = 1$). We focus on MLPs in which all internal functions $a^{(1)}, \ldots, a^{(k-1)}$ are the ReLU function $relu(x) = max(0, x)$. Usually, MLP binary classifiers are trained using the sigmoid as the output function $a^{(k)}$. Nevertheless, when an MLP classifies an input (after training), it takes decisions by simply using the preactivations, also called logits. Based on this and on the fact that we only consider already trained MLPs, we can assume w.l.o.g. that the output function $a^{(k)}$ is the binary *step* function, defined as $step(x) = 0$ if $x < 0$, and $step(x) = 1$ if $x \geq 0$.

A *perceptron* is an MLP with no hidden layers (i.e., $k = 1$). That is, a perceptron $\mathcal{M}$ is defined by a pair $(\mathbf{W}, b)$ such that $\mathbf{W} \in \mathbb{Q}^{n \times 1}$ and $b \in \mathbb{Q}$, and the output is $\mathcal{M}(\mathbf{x}) = step(\mathbf{x}\,\mathbf{W} + b)$. Because of its particular structure, a perceptron is usually defined as a pair $(\mathbf{w}, b)$ with $\mathbf{w} = \mathbf{W}^{\mathbf{T}}$ a rational vector and b a rational number. The output of $\mathcal{M}(\mathbf{x})$ is then 1 iff $\mathbf{x} \cdot \mathbf{w} + b \geq 0$, where $\mathbf{x} \cdot \mathbf{w}$ denotes the dot product between $\mathbf{x}$ and $\mathbf{w}$.

Boolean functions $\mathcal{F}$ mapping strings to strings whose output is a single bit are called decision problems. We identify the computational problem of computing $\mathcal{F}$ (i.e., given an input string x compute $\mathcal{F}(x)$) with the problem of deciding whether $\mathcal{F}(x) = 1$.

Even-if Explanations

In this section, we instantiate our framework on three important classes of boolean models and explainability queries. Subsequently, we present our main theorems, facilitating a comparison of these models in terms of their interpretability. We start by recalling the notion of counterfactual, that is the explainability notion in the 'if only' case, whose complexity has been investigated in (Barceló et al. 2020). We define the *distance measure* between two instances $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$ as the number of features where they differ. Formally, $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$ is the number of indexes $i \in \{1, \ldots, n\}$ (i.e., features) where $\mathbf{x}$ and $\mathbf{y}$ differ.

Definition 1 (Counterfactual)**.** *Given a pre-trained model $\mathcal{M}$ and an instance $\mathbf{x}$, an instance $\mathbf{y}$ is said to be a counterfactual of $\mathbf{x}$ iff i) $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{y})$, and ii) there exists no other instance $\mathbf{z} \neq \mathbf{y}$ s.t. $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{z})$ and $d(\mathbf{x}, \mathbf{z}) < d(\mathbf{x}, \mathbf{y})$.*

Example 3. Continuing with our running example illustrated in Figure 1, for $\mathbf{y}_3 = [1, 0, 1]$ we have that $\mathbf{x}_2 = [0, 0, 1]$ and $\mathbf{x}_3 = [1, 1, 1]$ are the only counterfactuals of $\mathbf{y}_3$ w.r.t. $\mathcal{M}$ (herein, $d(\mathbf{y}_3, \mathbf{x}_2) = d(\mathbf{y}_3, \mathbf{x}_3) = 1$). Intuitively, this encodes the fact that user $\mathbf{y}_3$ (that applied for a part-time and remote job, and a salary greater than or equal to 5K$) will be hired *if only* (s)he would change the employment contract to be full time (obtaining $\mathbf{x}_2$) or the requested salary to be lower than 5K$ (obtaining $\mathbf{x}_3$). □

The natural decision version of the problem of finding a couterfactual for $\mathbf{x}$ is the following.

Problem 1 ((Barceló et al. 2020))**.** [MINIMUM CHANGE REQUIRED (MCR)] *Given a model $\mathcal{M}$, instance $\mathbf{x}$, and $k \in \mathbb{N}$, check whether there exists an instance $\mathbf{y}$ with $d(\mathbf{x}, \mathbf{y}) \leq k$ and $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{y})$.*

Theorem 1 ((Barceló et al. 2020))**.** MCR *is i) in PTIME for FBDDs and perceptrons, and ii) NP-complete for MLPs.*

We follow a standard assumption about the relationship between model interpretability and computational complexity (Barceló et al. 2020): *a class $\mathcal{A}$ of models is more interpretable than another class $\mathcal{B}$ if the computational complexity of addressing post-hoc queries for models in $\mathcal{B}$ is higher than for those in $\mathcal{A}$.* Under this assumption, Theorem 1 states that the class of models 'perceptron' and 'FBDD' is strictly more interpretable than the class 'MLP', as the computational complexity of answering post-hoc queries for models in the first two classes is lower than for those in the latter. These results represent a principled way to confirm the folklore belief that linear models are more interpretable than deep neural networks within the context of interpretability queries for counterfactuals.

An open question is whether the same holds when dealing with post-hoc queries based on the 'even-if' thinking setting, i.e. on semifactuals. Before exploring this research question, we formally introduce the concept of semifactual.

Definition 2 (Semifactual)**.** *Given a pre-trained model $\mathcal{M}$ and an instance $\mathbf{x}$, an instance $\mathbf{y}$ is said to be a semifactual of $\mathbf{x}$ iff i) $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$, and ii) there exists no other instance $\mathbf{z} \neq \mathbf{y}$ s.t. $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{z})$ and $d(\mathbf{x}, \mathbf{z}) > d(\mathbf{x}, \mathbf{y})$.*

Similar to counterfactuals, the following problem is the decision version of the problem of finding a semifactual of an instance $\mathbf{x}$ with a model $\mathcal{M}$.

Problem 2. [MAXIMUM CHANGE ALLOWED (MCA)] *Given a model $\mathcal{M}$, instance $\mathbf{x}$, and $k \in \mathbb{N}$, check whether there is an instance $\mathbf{y}$ with $d(\mathbf{x}, \mathbf{y}) \geq k$ and $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$.*

Although semifactuals and counterfactuals appear to be similar, their mathematical definitions are different. Indeed, while counterfactuals minimize the changes in order to have a different outcome, semifactuals maximize the changes while keeping the same outcome. Notably, the two problems are not interchangeable - we do not see how to naturally reduce one to the other; however, a (possibly complex) reduction may exist as our complexity results presented in Theorem 2 below do not rule this out. For instance, considering Example 1 and the two semifactuals $\mathbf{y}_1$ and $\mathbf{y}_2$ of $\mathbf{x}_1$, they do not correspond to the counterfactuals of the counterfactuals of $\mathbf{x}_1$, that are $[0, 0, 1]$ and $[1, 1, 1]$.

Theorem 2. MCA *is i) in PTIME for FBDDs and perceptrons, and ii) NP-complete for MLPs.*

Proof sketch. (Perceptron). The MCA (optimization) problem under perceptrons can be formulated in ILP as the standard (max-) Knapsack problem where each item has value 1 and weight $w_i(2x_i - 1)$, and the bag has capacity $\sum_{i=1}^{n} w_i x_i$. The Knapsack problem is, in the general case, NP-Hard (Papadimitriou 1994). However, MCA corresponds to a special instance of Knapsack where every item has the same cost that can be solved in polynomial time with a greedy strategy, from which the result follows.
(FBDD). Let $\mathcal{M}_u$ be the FBDD obtained by restricting $\mathcal{M}$ to the nodes that are (forward-)reachable from u and $mca_u(\mathbf{x}) = \max\{k' \mid \exists \mathbf{y}.\ d(\mathbf{x}, \mathbf{y}) = k' \wedge \mathcal{M}_u(\mathbf{y}) = \mathcal{M}(\mathbf{x})\}$ (that can be comptuted in PTIME). For $\mathbf{y}$ maximizing k' it holds that $y_{u'} \neq x_{u'}$ holds $\forall u'$ from the root of $\mathcal{M}$ to u excluded. Let r be the root of $\mathcal{M}$. Then, we can show that $(\mathcal{M}, \mathbf{x}, k)$ is a positive instance of MCA iff $mca_r(\mathbf{x}) \geq k$.
(MLP) The following algorithm provides the membership in NP. Guess an instance $\mathbf{y}$ and check in PTIME that $d(\mathbf{x}, \mathbf{y}) \geq k$ and $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$. We prove the hardness with a polynomial reduction from INDEPENDENT SET, which is known to be NP-complete (Papadimitriou 1994). □

It turns out that, under standard complexity assumptions, computing semifactuals under perceptrons and FBDDs is easier than under multi-layer perceptrons. Moreover, independently of the type of the model, computing semifactuals is as hard as computing counterfactuals (cf. Table 1). Thus, perceptrons and FBDDs are strictly more interpretable than MLPs, in the sense that the complexity of answering post-hoc queries for models in the first two classes is lower than for those in the latter.

We point out that a limitation in using cunterfactuals and semifactuals may arise when the process of changing decision-outcomes is not 'monotonic', in the sense that after having obtained a decision change, by changing further feature values we obtain a further change to the decision-outcome, making it equal to the original value (e.g. a classification task establishing whether the number of bits is even). In such cases counterfactuals (resp. semifactuals), obtained by changing a set of features s, make sense only if there is no semifactual (resp. counterfactual) which can be obtained by changing a strict superset (resp. subset) of features. Thus, semifactuals are relevant even for understanding whether the computations of counterfactuals could be problematic and vice versa.

	FBDDs	PERCEPTRONS	MLPs
MCR	PTIME	PTIME	NP-c
MCA	PTIME	PTIME	NP-c
CB-MCR	coNP	coNP	coNP-c
CB-MCA	coNP	coNP	coNP-c
CBL-MCR	PTIME	PTIME	coNP-c
CBL-MCA	PTIME	PTIME	coNP-c

Table 1: Complexity of explainability queries for models of the form $\{0,1\}^n \to \{0,1\}$. For any class C, C-c means C-complete. Grey-colored cells refer to existing results.

Preferences over Explanations

The problem of preference handling has been extensively studied in AI. Several formalisms have been proposed to express and reason with different kinds of preferences (Brafman and Domshlak 2009; Rossi, Venable, and Walsh 2011; Santhanam, Basu, and Honavar 2016; Alfano et al. 2022, 2023a,b). In the following, in order to express preferences over semifactuals and counterfactuals, we introduce a novel approach inspired to that proposed in (Brewka, Niemelä, and Truszczynski 2003), whose semantics is based on the *degree* to which preference rules are satisfied.

Syntax. Input instances are of the form $\mathbf{x} = (x_1, ..., x_n)$ in $\{0,1\}^n$. Each x_i (with $i \in [1, n]$) represents the value of feature f_i, and the (equality) atom $f_i = x_i$ denotes that the value of feature f_i in $\mathbf{x}$ is equal to x_i. A *simple preference* is an expression of the form $(f_i = x_i') \succ (f_j = x_j'')$; it intuitively states that we prefer instances (i.e. semifactuals or counterfactuals) where $f_i = x_i'$ w.r.t. those where $f_j = x_j''$. To simplify the notation, an equality atom of the form $f_i = 1$ is written as a positive atom f_i, whereas an atom of the form $f_i = 0$ is written as a negated atom $\neg f_i$. Positive and negated atoms are also called (feature) literals.

Definition 3 (Preference rule)**.** Let $\mathcal{I} = \{f_1, ..., f_n\}$ be the set of input features. A preference rule is of the form:

$$\varphi_1 \succ \cdots \succ \varphi_k \leftarrow \varphi_{k+1} \wedge \cdots \wedge \varphi_m \qquad (1)$$

where $m \geq k \geq 2$, and any $\varphi_i \in \{f_1, \neg f_1, \ldots f_n, \neg f_n\}$ is a (feature) literal, with $i \in [1, m]$.

In (1), $\varphi_1 \succ \cdots \succ \varphi_k$ is called head (or consequent), whereas $\varphi_{k+1} \wedge \cdots \wedge \varphi_m$ is called body (or antecedent). We assume that literals in the head are distinct. Intuitively, whenever $\varphi_{k+1}, \cdots, \varphi_m$ are true, then φ_1 is preferred over φ_2, which is preferred over φ_3, and so on until φ_{k-1} which is preferred over φ_k. As usual, when the body of the rule is empty, the implication symbol $\leftarrow$ is omitted.

Definition 4 (BCMP framework)**.** A (binary classification) model with preferences (BCMP) framework is a pair $(\mathcal{M}, \succ)$ where $\mathcal{M}$ is a model and $\succ$[1] a set of preference rules over features of $\mathcal{M}$.

A practical and natural way for users to express their preferences in our framework includes—but are not limited to—specifying a ranking on a (sub)set of features whose values users would prefer to change (or keep unchanged) (Brewka, Niemelä, and Truszczynski 2003).

Example 4. The BCMP framework $\Lambda_1 = (\mathcal{M}, \{\neg f_2 \succ f_1\})$, where $\mathcal{M}$ is the model presented in Example 1, encodes the user's preference specified in Example 2. Consider now the framework Λ_2 obtained from Λ_1 by replacing the preference rule with $\neg f_2 \succ f_1 \leftarrow \neg f_3$. Roughly speaking, it encodes a user preference stating that among the explanations (i.e. counterfactuals/semifactuals) satisfying the condition that *the work is remote* (i.e., $\neg f_3$ holds), the ones where *the salary is greater than or equal to 5K$* (i.e., $\neg f_2$ holds) are preferred and, if this is not possible, those where *the work is part-time* (i.e., f_1) are taken. □

Semantics. To formally establish an ordering among explanations, a partial order $\sqsupseteq$ derived from the set of preference rules, is introduced next. Let us consider an explanation $\mathbf{y}$ and a preference κ of the form (1), the three situations are possible:

(a) the body of κ is not satisfied in $\mathbf{y}$;[2]
(b) the body of κ is satisfied in $\mathbf{y}$ and at least one head literal is true in $\mathbf{y}$;
(c) the body of κ is satisfied in $\mathbf{y}$ and none of the head literals is satisfied in $\mathbf{y}$.

In the cases (a) and (b) we say that $\mathbf{y}$ *satisfies* κ respectively with degree 1 and $min(\{l \mid \mathbf{y}$ satisfies φ_l with $l \leq k\})$ (denoted as $\delta(\mathbf{y}, \kappa)$), while in case (c) we say that $\mathbf{y}$ *does not satisfy* κ and the associated degree is $\delta(\mathbf{y}, \kappa) = +\infty$. Intuitively, $\delta(\mathbf{y}, \kappa)$ represents the position of the first feature literal satisfied in the ordered list provided in the head of a preference rule; however, it can be 1 if $\mathbf{y}$ satisfies the first literal φ_1 or if the rule is irrelevant (case a). If the rule is relevant (the body is satisfied) and no head literal is satisfied, then it is $+\infty$.

Thus, given a BCMP framework $(\mathcal{M}, \succ)$, an instance $\mathbf{x}$ and two explanations $\mathbf{y}$ and $\mathbf{z}$ for $\mathcal{M}$ and $\mathbf{x}$, we write $\mathbf{y} \sqsupseteq \mathbf{z}$ iff $\delta(\mathbf{y}, \kappa) \leq \delta(\mathbf{z}, \kappa)$ for all preferences κ in $\succ$, and write $\mathbf{y} \sqsupset \mathbf{z}$ iff $\mathbf{y} \sqsupseteq \mathbf{z}$ and $\mathbf{z} \not\sqsupseteq \mathbf{y}$.

Definition 5 (Semantics)**.** Given a BCMP framework $(\mathcal{M}, \succ)$ and an instance $\mathbf{x}$. We say that $\mathbf{y}$ is a *best semifactual* (resp., *counterfactual*) explanation of $\mathbf{x}$ if $\mathbf{y}$ is a semifactual (resp., counterfactual) of $\mathbf{x}$ and there is no other semifactual (resp., counterfactual) $\mathbf{z}$ of $\mathbf{x}$ such that $\mathbf{z} \sqsupset \mathbf{y}$.

Example 5. Continuing with Example 4, the instances $\mathbf{y}_1 = [0, 0, 0]$ and $\mathbf{y}_2 = [1, 1, 0]$ satisfy $\kappa = \neg f_2 \succ f_1 \leftarrow \neg f_3$ with degree $\delta(\mathbf{y}_1, \kappa) = 1$ and $\delta(\mathbf{y}_2, \kappa) = 2$ since the second and first literal in the head of κ is satisfied in $\mathbf{y}_1$ and $\mathbf{y}_2$, respectively, and the body of κ is satisfied by both of them. Then, $\mathbf{y}_1 \sqsupset \mathbf{y}_2$, and thus $\mathbf{y}_1$ is the only best semifactual of $\mathbf{x}_1 = [0, 1, 1]$. □

[1]With a little abuse of notation, we use $\succ$ to denote both a set of preferences and the preference relation among feature literals.

[2]A positive (resp., negative) atom f_i (resp., $\neg f_i$) is satisfied in $\mathbf{y}$=$(y_1, \ldots, y_n)$ whenever y_i=1 (resp., y_i=0).

Computational Complexity. We now investigate the complexity of several problems related to prioritized reasoning for explanations in order to compare model classes, even under prioritized reasoning. Observe that deciding the *existence of a best explanation* in the even-if and if-only thinking follows from deciding the existence of counterfactuals and semifactuals, respectively. Thus, preferences do not make the existence problem harder. Consider now the problem of *verification of a best explanation* in both settings.

Problem 3. [CHECK BEST-MCR (CB-MCR)] *Given a* BCMP $(\mathcal{M}, \succ)$*, instances* $\mathbf{x}$*,* $\mathbf{y}$ *with* $d(\mathbf{x}, \mathbf{y}) = k$*, and* $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{y})$*, check whether there is no* $\mathbf{z}$ *with* $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{z})$ *and either* $d(\mathbf{x}, \mathbf{z}) \leq k-1$*, or* $d(\mathbf{x}, \mathbf{z}) = k$ *and* $\mathbf{z} \sqsupset \mathbf{y}$.

Problem 4. [CHECK BEST-MCR (CB-MCA)] *Given a* BCMP $(\mathcal{M}, \succ)$*, instances* $\mathbf{x}$*,* $\mathbf{y}$ *with* $d(\mathbf{x}, \mathbf{y}) = k$*, and* $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{y})$*, check whether there is no* $\mathbf{z}$ *with* $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{z})$ *and either* $d(\mathbf{x}, \mathbf{z}) \geq k+1$ *or* $d(\mathbf{x}, \mathbf{z}) = k$ *and* $\mathbf{z} \sqsupset \mathbf{y}$.

In our semantics for any framework $(\mathcal{M}, \succ)$ and instances $\mathbf{y}, \mathbf{z}$ of $\mathcal{M}$, deciding whether $\mathbf{y} \sqsupset \mathbf{z}$ can be done in PTIME. We use this result to prove the following theorem.

Theorem 3. CB-MCR *and* CB-MCA *are* i*) in coNP for FBDDs and perceptrons, and* ii*) coNP-complete for MLPs.*

Proof sketch: Consider the CB-MCR problem. We show the membership in NP for its complement, denoted as $\overline{\text{CB-MCR}}$ (that is, checking whether $\mathbf{y}$ is *not* a best counterfactual of $\mathbf{x}$). Guess an instance $\mathbf{z}$ and check that $\mathcal{M}(\mathbf{x}) \neq \mathcal{M}(\mathbf{z})$, and either $d(\mathbf{x}, \mathbf{z}) < k$ or $d(\mathbf{x}, \mathbf{z}) = k$ and $\mathbf{z} \sqsupset \mathbf{y}$ holds. If the above condition holds then return NO, otherwise YES. As all conditions can be check in PTIME, $\overline{\text{CB-MCR}}$ is in NP, and thus CB-MCR is in coNP. The hardness result follows by letting $\succ = \emptyset$. As $\mathbf{z} \sqsupset \mathbf{y}$ is always false, the problem collapses to $\overline{\text{MCR}}$. The proof for CB-MCA is similar. □

Linear preferences. We now focus on a simpler yet expressive class of BCMPs where a fixed linear order is defined over a subset of features.

Definition 6 (BCMLP framework)**.** A (binary classification) model with linear preferences (BCMLP) framework is a pair $(\mathcal{M}, \succ)$ where $\mathcal{M}$ is a model, and $\succ$ consists of a linear preference rule, that is a single preference rule of the form (1) with empty body.

Example 6. Considering Example 4, the BCMP framework Λ_1 is linear, whereas Λ_2 is not. □

We use CBL-MCA and CBL-MCR to denote CB-MCA and CB-MCR where the input BCMP is linear. We will show that these problems are in PTIME in the case of perceptrons and FBDDs. The algorithms presented below take as input a model $\mathcal{M}$, an instance $\mathbf{x}$, and a linear preference κ, and returns a best semifactual explanation in PTIME. When a preference is not specified, a generic semifactual at maximum distance is returned. For the sake of the presentation, w.l.o.g., we assume that head literals are positive atoms (i.e., $\varphi_i = f_i$).

Algorithm 1: Computing a (best) semifactual for perceptrons

Input: Perceptron $\mathcal{M} = (\mathbf{W}, b)$, instance $\mathbf{x} \in \{0,1\}^n$, and linear preference $\kappa = f_{p_1} \succ \cdots \succ f_{p_l}$.
Output: A best semifactual $\mathbf{y}$ for $\mathbf{x}$ w.r.t. $\mathcal{M}$ and κ

1: Let $\mathbf{s} = [f_1/s_1, \ldots, f_n/s_n]$ where $\forall i \in [1,n]$, $s_i = 2x_iw_i - w_i$ if $\mathcal{M}(\mathbf{x}) = 1$, $w_i - 2x_iw_i$ otherwise;
2: Let $\mathbf{s}' = [f_{q_1}/s_{q_1}, ..., f_{q_n}/s_{q_n}]$ be the sorted version of $\mathbf{s}$ in ascending order of s_i;
3: $k = \max(\{i \in [0,n] \mid \mathcal{M}(\mathsf{flip}(\mathbf{x}, \mathsf{pos}(\mathbf{s}', i)) = \mathcal{M}(\mathbf{x})\})$;
4: **if** $k = 0$ **return** $\mathbf{x}$;
5: **if** $k = n$ **return** $[1 - x_1, \ldots, 1 - x_n]$;
6: $\mathbf{y} = \mathsf{flip}(\mathbf{x}, \mathsf{pos}(\mathbf{s}', k))$;
7: $\delta = \min(\{i \in [1,l] \mid y_{p_i} = 1\} \cup \{l+1\})$;
8: **for** $i \in [1, ..., \delta - 1]$ **do**
9: **if** $y_{p_i} = 1$ **return** $\mathbf{y}$;
10: Let $j = q_k$ if $x_{p_i} = y_{p_i}$, $j = q_{k+1}$ otherwise;
11: $\mathbf{z} = \mathsf{flip}(\mathbf{y}, \{p_i, j\})$;
12: **if** $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{z})$ **return** $\mathbf{z}$;
13: **return** $\mathbf{y}$;

Algorithm 2: Computing a (best) semifactual for FBDDs

Input: FBDD $\mathcal{M}$=$(V, E, \lambda_V, \lambda_E)$ with root t, instance $\mathbf{x} \in \{0,1\}^n$, and linear preference $\kappa = f_{p_1} \succ \cdots \succ f_{p_l}$.
Output: A best semifactual $\mathbf{y}$ for $\mathbf{x}$ w.r.t. $\mathcal{M}$ and κ.

1: Let $\mathcal{M}'$=$(V'$=V, E'=$E, \lambda_{V'}$=$\lambda_V, \lambda_{E'})$ be a copy of $\mathcal{M}$ where $\lambda_{E'}(u,v)$=1 if $(x_{\lambda_V(u)}$=$\lambda_E(u,v))$, 0 otherwise;
2: Let $\mathcal{N} = \mathsf{subgraph}(\mathcal{M}', \mathcal{M}(\mathbf{x}))$;
3: Let Π be the set of paths in $\mathcal{N}$ from t to leaf nodes;
4: **for** $f_{p_i} \in [f_{p_1}, \ldots, f_{p_l}]$ **do**
5: **if** $\exists \pi \in \Pi$ with $\mathbf{y} = \mathsf{build}(\mathbf{x}, \pi)$ and $y_{p_i} = 1$ **return** $\mathbf{y}$;
6: Let π be a path of Π taken non-deterministically;
7: **return** $\mathbf{y} = \mathsf{build}(\mathbf{x}, \pi)$;

Algorithm 1 is defined for the perceptron model. Initially, a list $\mathbf{s}$ of pairs feature/weight is built, where each weight takes into account the contribution of the associated feature to the result (Line 1). At Line 2 the list is sorted in ascending order of weights, giving a new list $\mathbf{s}'$. Next (Lines 3-6) the feature values of $\mathbf{x}$ are changed to get a semifactual instance $\mathbf{y}$. To change a maximum number k of feature values of $\mathbf{x}$, guaranteeing that the output of the model $\mathcal{M}(\mathbf{x})$ does not change, the order in $\mathbf{s}'$ is followed. To this end, the following functions are introduced: *i*) $\mathsf{pos}(\mathbf{s}', i)$, computing the set of the positions in $\mathbf{s}$ of the first i features in $\mathbf{s}'$, and *ii*) $\mathsf{flip}(\mathbf{y}, B)$, with $\mathbf{y} = [y_1, ..., y_n]$, updating every element y_i such that $i \in B$ with the complementary value $1 - y_i$. Notice that $k = 0$ means that $\mathbf{x}$ is the only semifactual for $\mathbf{x}$, (returned at Line 4), whereas $k = n$ means that $[1 - x_1, \ldots, 1 - x_n]$ is the only semifactual for $\mathbf{x}$ (returned at Line 5). At Line 7 the degree of satisfaction of κ by $\mathbf{y}$ is computed; if no feature in κ is satisfied by $\mathbf{y}$ the degree is $l + 1$ (standing for $+\infty$). The next steps (Lines 8-13) search for a better semifactual instance (if any). This is carried out by considering the first $\delta - 1$ features in κ. Thus, for each feature f_{p_i} in κ ($i \in [1, \delta - 1]$) we have that: *i*) if $\mathbf{y}$ satisfies f_{p_i}, $\mathbf{y}$ is a best semifactual and it is returned at Line 9; *ii*) otherwise an alternative instance $\mathbf{z}$ satisfying feature f_{p_i} is generated and, if it is a semifactual then it is returned at Line 12. In particular, $\mathbf{z}$ is a semifactual instance only if $d(\mathbf{z}, \mathbf{x}) = d(\mathbf{y}, \mathbf{x})$ and $\mathcal{M}(\mathbf{z}) = \mathcal{M}(\mathbf{x})$. To guarantee that $d(\mathbf{z}, \mathbf{x}) = d(\mathbf{y}, \mathbf{x})$ we either restore feature f_{q_k} in $\mathbf{y}$ by setting it back to $y_{q_k} = x_{q_k}$ whenever $x_{p_i} = y_{p_i}$ (recall that the first k features of $\mathbf{x}$ are flipped at Line 6), or change the feature f_{k+1} (i.e., $y_{k+1} \neq x_{k+1}$) whenever $x_{p_i} \neq y_{p_i}$. Roughly speaking, the so-obtained feature f_j minimally contributes to determine the value $\mathcal{M}(\mathbf{y})$ and keeps the distance equal to k. Finally, if there exist no other semifactual $\mathbf{z}$ for $\mathbf{x}$ s.t. $\mathbf{z} \sqsupset \mathbf{y}$, then $\mathbf{y}$ is returned at Line 13.

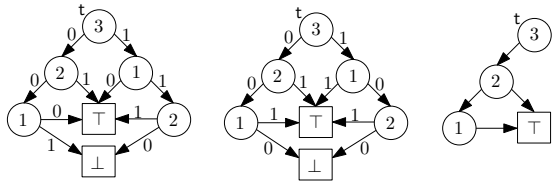

Figure 2: (Left) FBDD model $\mathcal{M} = (V, E, \lambda_V, \lambda_E)$ of Example 8 with root t and $\lambda_V(\mathsf{t}) = 3$. (Center) FBDD model $\mathcal{M}' = (V', E', \lambda_{V'}, \lambda_{E'})$ computed at Line 1 of Algorithm 2. (Right) Graph $\mathcal{N}$ obtained at Line 2 of Algorithm 2. Squared nodes represent leaf nodes ($\top/\bot$ for $\mathcal{M}(\cdot) = 1/0$).

Example 7. Consider the model $\mathcal{M} = step(\mathbf{x} \cdot [-2, 2, 0] + 1)$ of Example 1, the instance $\mathbf{x} = [0, 1, 1]$ and the linear preference $f_1 \succ f_2$. As $\mathcal{M}(\mathbf{x}) = 1$, we compute s_1 as $2x_1w_1 - w_1 = 2 \cdot 0 \cdot (-2) - (-2) = 2$. Analogously, $s_2 = 2x_2w_2 - w_2 = 2 \cdot 1 \cdot 2 - 2 = 2$ and $s_3 = 2x_3w_3 - w_3 = 2 \cdot 1 \cdot 0 - 0 = 0$. Then, $\mathbf{s} = [f_1/s_1, f_2/s_2, f_3/s_3] = [f_1/2, f_2/2, f_3/0]$, and its sorted version in ascending order of s_is is $\mathbf{s}' = [f_3/0, f_2/2, f_1/2]$. As $k = 2$, by flipping in $\mathbf{s}$ the features occurring in the first k position of $\mathbf{s}'$ we get $\mathbf{y} = [0, 0, 0]$ with $\delta = l + 1 = 3$ as no feature in the head of κ is true in $\mathbf{y}$. Finally, observing that $p_1 = 1$ and $p_2 = 2$, as $y_1 = x_1 = 0$, Algorithm 1 returns $\mathbf{z} = \mathsf{flip}(\mathbf{y}, [1, j]) = [1, 1, 0]$ with $j = q_k = 2$ and $\delta = 1$. □

Algorithm 2 works with FBDD. It starts by creating a copy $\mathcal{M}'$ of $\mathcal{M}$ where edge labels denote changes in the input features, that is $\lambda_{E'}(u, v) = 0$ (resp., $\lambda_{E'}(u, v) = 1$) denotes the fact that the value of feature $x_{\lambda_V(u)}$ is changed (resp., confirmed). Semifactuals are obtained by finding the paths π' in $\mathcal{M}'$ from the root node t to the leaf node representing $\mathcal{M}(\mathbf{x})$ (i.e., to $\top$ if $\mathcal{M}(\mathbf{x}) = 1$, $\bot$ otherwise) with a minimum number of 1's. The application of the changes occurring in a paths π' allow obtaining a semifactual for $\mathbf{x}$. Let n be the number of input features and let w be the number of 1's in the path π' to derive semifactual $\mathbf{y}$, $n - w$ denotes the distance $d(\mathbf{x}, \mathbf{y})$. It is worth noting that all paths π' have the same number of 1's, that is all semifactuals share the same distance. To this end, at Line 2 a graph $\mathcal{N}$ is built with function $\mathsf{subgraph}$ by keeping only nodes and edges in the paths of $\mathcal{M}'$ ending in $\mathcal{M}(\mathbf{x})$ and having minimum weight. All such paths are stored in Π at Line 3. Then, the algorithm checks if it is possible to build a semifactual $\mathbf{y}$ for $\mathbf{x}$ satisfying f_{p_1}, otherwise f_{p_2}, and so on. Particularly, assuming to be at step f_{p_i} for some f_{p_i} in κ, if there exists a path $\pi \in \Pi$ and the feature f_{p_i} can be set to 1 in $\mathbf{y}$ (that

is the condition of Line 5) then a best semifactual $\mathbf{y}$ of $\mathbf{x}$ is obtained from $\mathbf{x}$ by flipping every features i of $\mathbf{x}$ not appearing in π (i.e., $\lambda_V(u) \neq i$ for any node u in π) or differing in the assignment given by π (i.e., there is no edge $(u, v) \in \pi$ s.t. $\lambda_V(u) = i$ and $\lambda_{E'}(u, v) = 0$). More formally, the function $\mathsf{build}(\mathbf{x}, \pi)$ returns the instance $\mathbf{y} = \mathsf{flip}(\mathbf{x}, \{i \in [1, n] \mid \nexists (u, v) \in \pi$ such that $\lambda_V(u)$=i and $\lambda_{E'}(u, v)$=1$\})$. Finally, if the algorithm does not return a semifactual at Line 6, then at Line 8 it returns a semifactual $\mathbf{y}$ of $\mathbf{x}$ that satisfies none of the f_{p_i}s, obtained from $\mathbf{x}$ through function $\mathsf{build}(\mathbf{x}, \pi)$ where π is a path taken non-deterministically from Π.

Example 8. Consider the FBDD $\mathcal{M} = (V, E, \lambda_V, \lambda_E)$ in Figure 2 (left) for the hiring scenario of Example 1. Let $\mathbf{x} = [0, 1, 1]$ and $\kappa = f_2 \succ f_1$. For each edge $(u, v) \in E$, Figure 2 (center) shows the value $\lambda_{E'}(u, v)$ (computed at Line 1). For instance, $\lambda_{E'}(3, 1) = 1$ as $x_3 = \lambda_E(3, 1) = 1$, whereas $\lambda_{E'}(1, 2) = 0$ as $x_1 = 0 \neq \lambda_E(1, 2) = 1$. Figure 2 (right) shows the graph $\mathcal{N}$ obtained from $\mathcal{M}'$ by keeping only nodes and edges occurring in paths with a minimum weight (equal to 1). This means that semifactuals $\mathbf{y}$ of $\mathbf{x}$ are at distance $d(\mathbf{x}, \mathbf{y}) = n - 1 = 2$. As there is in $\mathcal{N}$ the path $\pi : (u, u'), (u', u'')$ where $\lambda_V(u) = 3$, $\lambda_V(u') = 2$, $\lambda_V(u'') = \top$, $\lambda_{E'}(u, u') = 0$, and $\lambda_{E'}(u', u'') = 1$ then we can get a semifactual $\mathbf{y}$ from $\mathbf{x}$ by changing all feature values apart from x_2, i.e. $\mathbf{y}$=$\mathsf{build}(\mathbf{x}, \pi)$=$\mathsf{flip}(\mathbf{x}, \{1, 3\})$=$[1, 1, 0]$. □

Theorem 4. CBL-MCA *and* CBL-MCR *are* i) *in PTIME for perceptrons and FBDDs, and* ii) *coNP-complete for MLPs.*

Providing a PTIME algorithm returning a best semifactual for MLPs is unfeasible, as backed by our complexity analysis. However, as an heuristic, we could adapt Algorithm 1 so that the scores s_is encode feature importance, similarly to what is done in (Ramon et al. 2020)—we plan to explore this direction in future work. Observe that, although the number of (best) semifactuals is potentially exponential w.r.t. the number of features (it is $\binom{n}{k}$ where k represents the maximum number of features changed in $\mathbf{x}$ to obtain semifactuals), Algorithms 1 and 2 can be exploited to obtain a finite representation of all semifactuals. For instance, all semifactuals of $\mathbf{x}$ in Algorithm 2 are those obtained from $\mathbf{x}$ by considering all paths in $\pi \in \Pi$, that is the set $\{\mathsf{build}(\mathbf{x}, \pi)$ such that $\pi \in \Pi\}$.

Related Work

Looking for transparent and interpretable models has led to the exploration of several explanation paradigms in eXplainable AI (XAI) (Marques-Silva and Ignatiev 2022; Ignatiev et al. 2022; Malfa et al. 2021; Ignatiev, Narodytska, and Marques-Silva 2019). Factual explanations (Ciravegna et al. 2020; Guidotti et al. 2018; Bodria et al. 2023; Wang, Khosravi, and den Broeck 2021; Ciravegna et al. 2023; Cooper and Marques-Silva 2023) elucidate the inner workings of AI models by delineating why a certain prediction was made based on the input data. Counterfactual explanations (Dervakos et al. 2023; Romashov et al. 2022; Albini et al. 2020; Wu, Zhang, and Wu 2019; Guidotti 2022; Audemard et al. 2024a,b), delve into hypothetical scenarios, revealing alternative input configurations that could have resulted in a different prediction. These explanations offer insights into the model's decision boundaries and aid in understanding its behavior under varied circumstances. Semifactual explanations (Dandl et al. 2023; Aryal and Keane 2023; Kenny and Keane 2021; Alfano et al. 2024b) bridge the gap between factual and counterfactual realms by presenting feasible alterations to the input data that do not change a decision outcome. This trichotomy of explanation types contributes significantly to the holistic comprehension and trustworthiness of AI systems, catering to various stakeholders' needs for transparency and interpretability. Most existing works on XAI focus on proposing novel methods for generating explanations, with few addressing the computational complexity of related problems (Barceló et al. 2020; Arenas et al. 2022, 2021; Eiben et al. 2023; El Harzli, Grau, and Horrocks 2023; Marzari et al. 2023; Ordyniak, Paesani, and Szeider 2023). However, none of these works on theoretical underpinnings of XAI specifically focuses on semifactuals. The approach proposed in (Dandl et al. 2023) aims to derive a set of semifactuals by solving an NP-hard optimization problem, for which they introduce a model-agnostic *heuristic* method. However, our approach differs from that in (Dandl et al. 2023) from several standpoints: (*i*) the notion of semifactual considered (that does not rely on maximal distance in that work), (*ii*) the ability to express preferences (not considered in that work), and, more importantly, (*iii*) the fact that our approach is an *exact* method rather than heuristic. Finally, (Darwiche and Hirth 2023) studies a different yet related problem, called even-if-because problem, for ordered binary decision diagrams.

Conclusions and Future Work

After exploring local post-hoc interpretability queries related to semifactuals and their complexity w.r.t. three classes of models, we have introduced a framework that enables users to personalize semifactual and counterfactual explanations based on preferences. Then, we investigated the complexity of this framework and presented PTIME algorithms.

Our work is also motivated by the growing interest in regulating AI (Buiten 2019), and particularly in contexts where sensitive decisions regarding humans are demanded to AI systems. An important example is given by the legislation proposal in the State of New York concerning the use of such systems in recruitment processes. The proposal emphasizes the need to explain systems' results, focusing not so much on the decision-making process itself, but rather on the outcome produced (Lohr 2023). As this is exactly the explainability setting considered in our work, we believe that our research could inspire future developments in regulating AI.

We plan to explore additional interpretability queries by considering e.g. i) counting problems like quantifying the number of semifactuals/counterfactuals, and ii) the recent notion of *alterfactual* explanations (Mertes et al. 2024), whose idea is to show irrelevant attributes of a predicted instance. Furthermore, we aim to extend our research by investigating more inclusive model formats, specifically those accommodating non-binary features. Finally, our algorithms focus on (i) Perceptrons and FBDD, and (ii) a single linear preference, though they can be easily adapted to cope with any arbitrary but fixed number of (consistent) linear preferences. Devising algorithms dealing with (i) MLP and (ii) multiple and different preference criteria deserves further investigation.

Acknowledgments

We acknowledge financial support from PNRR MUR projects FAIR (PE0000013) and SERICS (PE00000014), project Tech4You (ECS0000009), and MUR project PRIN 2022 EPICA (H53D23003660006).

References

Albini, E.; Rago, A.; Baroni, P.; and Toni, F. 2020. Relation-Based Counterfactual Explanations for Bayesian Network Classifiers. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 451–457.

Alfano, G.; Greco, S.; Mandaglio, D.; Parisi, F.; Shahbazian, R.; and Trubitsyna, I. 2024a. Even-if Explanations: Formal Foundations, Priorities and Complexity. *CoRR*, abs/2401.10938.

Alfano, G.; Greco, S.; Parisi, F.; and Trubitsyna, I. 2022. On Preferences and Priority Rules in Abstract Argumentation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2517–2524.

Alfano, G.; Greco, S.; Parisi, F.; and Trubitsyna, I. 2023a. Abstract Argumentation Framework with Conditional Preferences. In *Proceedings of AAAI Conference on Artificial Intelligence*, 6218–6227.

Alfano, G.; Greco, S.; Parisi, F.; and Trubitsyna, I. 2023b. Preferences and Constraints in Abstract Argumentation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3095–3103.

Alfano, G.; Greco, S.; Parisi, F.; and Trubitsyna, I. 2024b. Counterfactual and Semifactual Explanations in Abstract Argumentation: Formal Foundations, Complexity and Computation. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 14–26.

Arenas, M.; Baez, D.; Barceló, P.; Pérez, J.; and Subercaseaux, B. 2021. Foundations of symbolic languages for model interpretability. *Proceedings of Advances in Neural Information Processing Systems*, 34: 11690–11701.

Arenas, M.; Barceló, P.; Romero Orth, M.; and Subercaseaux, B. 2022. On computing probabilistic explanations for decision trees. *Proceedings of Advances in Neural Information Processing Systems*, 35: 28695–28707.

Aryal, S.; and Keane, M. T. 2023. Even If Explanations: Prior Work, Desiderata & Benchmarks for Semi-Factual XAI. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 6526–6535.

Audemard, G.; Lagniez, J.-M.; Marquis, P.; and Szczepanski, N. 2024a. Deriving Explanations for Decision Trees: The Impact of Domain Theories. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3688–3696.

Audemard, G.; Lagniez, J.-M.; Marquis, P.; and Szczepanski, N. 2024b. On The Computation of Example-Based Abductive Explanations for Random Forests. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3679–3687.

Bakker, M.; Chadwick, M.; Sheahan, H.; Tessler, M.; Campbell-Gillingham, L.; Balaguer, J.; McAleese, N.; Glaese, A.; Aslanides, J.; Botvinick, M.; et al. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *Proceedings of Advances in Neural Information Processing Systems*, 35: 38176–38189.

Barceló, P.; Monet, M.; Pérez, J.; and Subercaseaux, B. 2020. Model Interpretability through the lens of Computational Complexity. In *Proceedings of Advances in Neural Information Processing Systems*.

Bodria, F.; Giannotti, F.; Guidotti, R.; Naretto, F.; Pedreschi, D.; and Rinzivillo, S. 2023. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 1–60.

Brafman, R. I.; and Domshlak, C. 2009. Preference Handling - An Introductory Tutorial. *AI Mag.*, 30(1): 58–86.

Brewka, G.; Niemelä, I.; and Truszczynski, M. 2003. Answer Set Optimization. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 867–872.

Buiten, M. C. 2019. Towards intelligent regulation of artificial intelligence. *European Journal of Risk Regulation*, 10(1): 41–59.

Ciravegna, G.; Barbiero, P.; Giannini, F.; Gori, M.; Lió, P.; Maggini, M.; and Melacci, S. 2023. Logic explained networks. *Artificial Intelligence*, 314: 103822.

Ciravegna, G.; Giannini, F.; Gori, M.; Maggini, M.; and Melacci, S. 2020. Human-Driven FOL Explanations of Deep Learning. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2234–2240.

Cooper, M. C.; and Marques-Silva, J. 2023. Tractability of explaining classifier decisions. *Artif. Intell.*, 316: 103841.

Dandl, S.; Casalicchio, G.; Bischl, B.; and Bothmann, L. 2023. Interpretable Regional Descriptors: Hyperbox-Based Local Explanations. In *Proceedings of Machine Learning and Knowledge Discovery in Databases*, volume 14171, 479–495. Springer.

Darwiche, A.; and Hirth, A. 2023. On the (Complete) Reasons Behind Decisions. *J. Log. Lang. Inf.*, 32(1): 63–88.

Dervakos, E.; Thomas, K.; Filandrianos, G.; and Stamou, G. 2023. Choose your Data Wisely: A Framework for Semantic Counterfactuals. *arXiv preprint arXiv:2305.17667*.

Eiben, E.; Ordyniak, S.; Paesani, G.; and Szeider, S. 2023. Learning Small Decision Trees with Large Domain. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3184–3192.

El Harzli, O.; Grau, B. C.; and Horrocks, I. 2023. Cardinality-minimal explanations for monotonic neural networks. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3677–3685.

Guidotti, R. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 1–55.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5): 1–42.

Ignatiev, A.; Izza, Y.; Stuckey, P. J.; and Marques-Silva, J. 2022. Using MaxSAT for Efficient Explanations of Tree Ensembles. In *Proceedings of AAAI Conference on Artificial Intelligence*, 3776–3785.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019. Abduction-Based Explanations for Machine Learning Models. In *Proceedings of AAAI Conference on Artificial Intelligence*, 1511–1519.

Jiang, J.; Leofante, F.; Rago, A.; and Toni, F. 2024. Robust counterfactual explanations in machine learning: A survey. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 8086–8094.

Kahneman, D.; and Tversky, A. 1981. *The simulation heuristic*. National Technical Information Service.

Kapoor, A.; Lee, B.; Tan, D.; and Horvitz, E. 2012. Performance and preferences: Interactive refinement of machine learning procedures. In *Proceedings of AAAI Conference on Artificial Intelligence*, 1578–1584.

Kenny, E. M.; and Huang, W. 2023. The Utility of "Even if" Semi-Factual Explanation to Optimise Positive Outcomes. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kenny, E. M.; and Keane, M. T. 2021. On generating plausible counterfactual and semi-factual explanations for deep learning. In *Proceedings of AAAI Conference on Artificial Intelligence*, 11575–11585.

Lohr, S. 2023. A Hiring Law Blazes a Path for A.I. Regulation. New York Times.

Malfa, E. L.; Michelmore, R.; Zbrzezny, A. M.; Paoletti, N.; and Kwiatkowska, M. 2021. On Guaranteed Optimal Robust Explanations for NLP Models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2658–2665.

Marques-Silva, J.; and Ignatiev, A. 2022. Delivering Trustworthy AI through Formal XAI. In *Proceedings of AAAI Conference on Artificial Intelligence*, 12342–12350.

Marzari, L.; Corsi, D.; Cicalese, F.; and Farinelli, A. 2023. The #DNN-Verification Problem: Counting Unsafe Inputs for Deep Neural Networks. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 217–224.

McCloy, R.; and Byrne, R. M. 2002. Semifactual "even if" thinking. *Thinking & reasoning*, 8: 41–67.

Mertes, S.; Huber, T.; Karle, C.; Weitz, K.; Schlagowski, R.; Conati, C.; and André, E. 2024. Relevant Irrelevance: Generating Alterfactual Explanations for Image Classifiers. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 467–475.

Ordyniak, S.; Paesani, G.; and Szeider, S. 2023. The Parameterized Complexity of Finding Concise Local Explanations. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3312–3320.

Papadimitriou, C. H. 1994. *Computational complexity.* Addison-Wesley.

Ramon, Y.; Martens, D.; Provost, F.; and Evgeniou, T. 2020. A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. *Advances in Data Analysis and Classification*, 14: 801–819.

Romashov, P.; Gjoreski, M.; Sokol, K.; Martinez, M. V.; and Langheinrich, M. 2022. BayCon: Model-agnostic Bayesian Counterfactual Generator. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 23–29.

Rossi, F.; Venable, K. B.; and Walsh, T. 2011. *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Santhanam, G. R.; Basu, S.; and Honavar, V. G. 2016. *Representing and Reasoning with Qualitative Preferences: Tools and Applications*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Wang, E.; Khosravi, P.; and den Broeck, G. V. 2021. Probabilistic Sufficient Explanations. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 3082–3088.

Wegener, I. 2004. BDDs—design, analysis, complexity, and applications. *Discrete Applied Mathematics*, 138(1): 229–251.

Wu, M.; Wu, H.; and Barrett, C. W. 2023. VeriX: Towards Verified Explainability of Deep Neural Networks. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proceedings of Advances in Neural Information Processing Systems*.

Wu, Y.; Zhang, L.; and Wu, X. 2019. Counterfactual fairness: Unidentification, bound and algorithm. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 1438–1444.

Zhu, Y.; Tang, Z.; Liu, Y.; Zhuang, F.; Xie, R.; Zhang, X.; Lin, L.; and He, Q. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1507–1515.