A Unified Logical Framework for Explanations in Classifier Systems

Xinghan Liu, Emiliano Lorini

A Unified Logical Framework for Explanations in Classifier Systems

Xinghan Liu[1] and Emiliano Lorini[2]
[1]IRIT, Toulouse University, France xinghan.liu@univ-toulouse.fr
[2]IRIT-CNRS, Toulouse University, France Emiliano.Lorini@irit.fr

Abstract

Recent years have witnessed a renewed interest in the explanation of classifier systems in the field of explainable AI (XAI). The standard approach is based on propositional logic. We present a modal language which supports reasoning about binary input classifiers and their properties. We study a family of classifier models, axiomatize it as two proof systems regarding the cardinality of the language and show completeness of our axiomatics. Moreover, we show that the satisfiability checking problem for our modal language is NEXPTIME-complete in the infinite-variable case, while it becomes polynomial in the finite-variable case. We moreover identify an interesting NP fragment of our language in the infinite-variable case. We leverage the language to formalize counterfactual conditional as well as a variety of notions of explanation including abductive, contrastive and counterfactual explanations, and biases. Finally, we present two extensions of our language: a dynamic extension by the notion of assignment enabling classifier change and an epistemic extension in which the classifier's uncertainty about the actual input can be represented.

1 Introduction

The notions of explanation and explainability have been extensively investigated by philosophers [20, 25, 50] and are key aspects of AI-based systems given the importance of explaining the behavior and prediction of an artificial intelligent system. Classifier systems compute a given function in the context of a classification or prediction task. Artificial feedforward neural networks are special kinds of classifier systems aimed at learning or, at least approximating, the function mapping instances of the input data to their corresponding outputs. Explaining why a system has classified a given instance in a certain way is crucial for making the system intelligible and for finding biases in the classification process. This is the main target of explainable AI (XAI). Thus, a variety of notions have been defined and used to explain classifiers including abductive, contrastive and counterfactual explanations [4, 49, 13, 24, 36, 34, 37, 48, 35, 33].

Inputs of a classifier are called instances, i.e., valuations of all its variables/features/factors, and outputs are called classifications/predictions/decisions.[1] When both input and output of the classifier are binary, it is just a Boolean function $f : \{0,1\}^n \longrightarrow \{0,1\}$, and furthermore can be expressed by a propositional formula. This isomorphism between Boolean functions and logic has been known ever since the seminal work of Boole. Recently there has been a renewed interest in Boolean functions in the area of logic-based approaches to XAI [41, 24, 12, 23, 40, 2, 1]. They concentrate on *local* explanations, i.e., on explaining why an actual instance is classified in a certain way.

We argue that it is natural and fruitful to represent binary (input) classifiers and their explanations with the help of a modal language. To that end let us first explain the conceptual foundation of explanation in the context of classifiers, which is largely ignored in the recent literature.

What is an explanation? Despite subtle philosophical debates,[2] by explanation people usually mean *causal explanation*, an answer to a "why" question in terms of "because". Then what is a causal explanation? Ever since the seminal deductive-nomological (D-N) model [20], one can view it as a logical relation between an explanandum (the proposition being explained) and an explanans (the proposition explaining), which is itself expressible by a logical formula. According to the D-N model, a causal explanation of a certain fact should include a reference to the *laws* that are used for deducing it from a set of premises.

More recently Woodward & Hitchcock [52, p. 2, p. 17] (see also [51, Ch. 5 and 6]) proposed that causal explanations make reference to generalizations, or descriptions of dependency relations, which specify relationships between the explanans and explanandum variables. No need of being laws, such generalizations exhibit how the explanandum variable is counterfactually dependent on the explanans variables by relating changes in the value of the latter to changes in the value of the former.[3] According to Woodward & Hitchcock, a generalization used in a causal explanation is *invariant under intervention* insofar as it remains stable after changing the actual value of the variables cited in the explanation.[4]

We claim that existing notions of explanation leveraged in the XAI domain rest upon the idea of invariance under intervention. However, while Woodward & Hitchcock apply it to the notion of generalization, in the XAI domain it usually concerns the result of the classifier's decision to be explained. Another minor difference with Woodward & Hitchcock is terminological: when explaining

[1] We use them as synonyms through the paper. Another set of synonyms is perturbation/intervention/manipulation. The variety of terminology is unfortunate.

[2] E.g., whether all explanations are causal, whether metaphysical explanation/grounding should be distinguished from causal explanation.

[3] Using the notion of counterfactual dependence for reasoning about natural laws and causality traces back to [16, 28, 30]. The focus nowadays, e.g. [50, 19], is on the use of counterfactuals for modeling the notion of *actual* cause in order to test (rather than define) causality.

[4] Woodward & Hitchcock also discuss invariance with respect to the background conditions not figuring in the relationship between explanans and explanandum. Nonetheless, they consider this type of invariance less central to causal explanation.

the decision of a binary classifier system, the term 'perturbation' is commonly used instead of 'intervention'. But they both mean switching some features' values from the current ones to other ones. Let us outline it by introducing informally our running example.

Example 1 (Applicant Alice, informal)**.** *Alice applies for a loan. She is not male, she is employed, and she rents an apartment in the city center, which we note* $\neg male \wedge employed \wedge \neg owner \wedge center$*. The classifier* f *only accepts the application if the applicant is employed, and either is a male or owns a property. Hence, Alice's application is rejected.*

In the XAI literature $\neg male \wedge \neg owner$ is called an abductive explanation (AXp) [24] or sufficient reason [12] of the actual decision of rejecting Alice's application, because perturbing the values of the other features ('employment' and 'address' in this setting), while keeping the values of 'gender' or 'ownership' fixed, will not change the decision. More generally, for a term (a conjunction of literals) to be an abductive explanation of the classifier's actual decision, the classifier's decision should be invariant under perturbation on the variables not appearing in the term.[5]

On the contrary, $\neg male$ is called a contrastive explanation (CXp) [23],[6] because perturbing nothing but 'gender' will change the decision from rejecting the application to accepting it. Therefore, the "duality" between two notions rests on the fact that AXp answers a *why*-question by indicating that the classification would stay unchanged under intervention on variables other than 'gender' and 'ownership', whereas CXp answers a *why not*-question by indicating that the classification would change under intervention on 'gender'. More generally, for a term (a conjunction of literals) to be a contrastive explanation of the classifier's actual decision, the classifier's decision should be variant under perturbation on *all* variables appearing in the term, where 'variant' is assumed to be synonym of 'non-invariant'.

As Woodward [50, p. 225, footnote 5] clarifies:

> [I]nvariance is a *modal* notion – it has to do with whether a relationship would remain stable under various hypothetical changes.

Therefore, following Woodward, the most natural way of modeling invariance is by means of a modal language whereby the notions of necessity and possibility can be represented. This is the approach we take in this work.

In particular, in order to model explanations in classifier systems, we use a modal language with a *ceteris paribus* (other things being equal) flavor. Indeed, the notion of invariance under intervention we consider presupposes that one intervenes on specific input features of the classifier, while keeping the values of the other input features unchanged (i.e., the values of the other input features being equal). So, for Alice's example we expect two modal formulas saying:

[5] AXp satisfies an additional restriction of minimality that will be elucidated at a later stage: an AXp is a 'minimal' term for which the classifier's actual decision is invariant under perturbation.

[6] We prefer the notation AXp used by Ignatiev et al.[24, 23] for its connection with CXp.

a) 'gender' and 'ownership' keeping their actual values, changing other features' values *necessarily* does not affect the actual decision of rejecting Alice's application;

b) other features keeping their actual values, changing the value of 'gender' *necessarily* modifies the classifier's decision of rejecting Alice's application.

Specifically, we will extend the *ceteris paribus* modal logic introduced in [17] by a finite set of atoms representing possible decisions/classifications of a classifier and axioms regarding them. The resulting logic is called BCL which stands for Binary input Classifier Logic, since the input variables of a classifier are assumed to be binary. One may roughly thinks of its models as S5 models supplemented with a classification function which allows us to fully represent a classifier system. Each state in the model corresponds to a possible input instance of the classifier. Moreover, the classification function induces a partition of the set of instances, where each part corresponds to a set of input instances which are classified equally by the classifier. We call these models *classifier models*. BCL and its extensions open up new vistas including (i) defining counterfactual conditionals and studying their relationship with the notions of abductive and contrastive explanation, (ii) modeling classifier dynamics through the use of formal semantics for logics of communication and change [43, 46], and (iii) viewing a classifier as an agent and representing its uncertainty about the actual instance to be classified through the use of epistemic logic [14].

Before concluding this introduction, it is worth noting that a classifier system is a simple form of causal system whose only dependency relations are between the input variables and the single output variable. Unlike Bayesian networks or artificial neural networks, a classifier system does not include 'intermediate' endogenous variables that, at the same time, depend on the input variables and causally influence the output variable(s). Therefore, many distinctions and disputations addressed in the theory of causality and causal explanation do not emerge in our work. For example, the vital distinction between correlation and causality [38], the criticism of *ceteris paribus* as natural law [50], and whether a causal explanation requires providing information about a causal history or causal chain of events [29]. All these subtleties only show up when the causal structure is complex, and hence collapse in a classifier system, which has only two layers (input-output).

Outline The paper is structured as follows. In Section 2 we introduce our modal language as well as its formal semantics using the notion of classifier model. In Section 3 two proof systems are given, BCL and 'weak' BCL (WBCL). We show they are sound and complete relative to the classifier system semantics with, respectively, finite-input and infinite-input variables. Section 4 presents a family of counterfactual conditional operators and elucidates their relevance for understanding the behavior of a classifier system. Section 5 is devoted to classifier explanation. We extend the existing notions of explanation for Boolean classifiers to binary input classifiers. The notions include AXp, CXp and bias

in the field of XAI. We will see that in the binary input classifier setting their behaviors are subtler. Besides, their connection with counterfactual is studied. Finally, in Section 6 we present two extensions of our language: (i) a dynamic extension by the notion of assignment enabling classifier change and (ii) an epistemic extension in which the classifier's uncertainty about the actual input can be represented. Further possible researches are discussed in the conclusion. Main results are either proven in the appendix or pointed out as corollaries.

Compared to our conference paper presented at CLAR 2021 [31], we do not restrict anymore to the language with finite variables. Thus two proof systems instead of one have to be presented, because in the infinite-variable setting the "functionality" property of classifiers cannot be syntactically expressed in a finitary way. The assumption of complete domain is dropped partially for the same technical reason. As a result, we are able to represent partial classifiers which were not expressible in the framework presented in our CLAR 2021 paper. Completeness and complexity results have been refined and improved. Other parts also changed according to possibly infinite variables and incomplete domain.

2 A Language for Binary Classifiers

In this section we introduce a language for modeling binary (input) classifiers and its semantics. The language has a *ceteris paribus* nature that comes from the *ceteris paribus* operators of the form $[X]$ it contains. They were first introduced in [17].[7]

2.1 Basic Language and Classifier Model

Let Atm_0 be a countable set of atomic propositions with elements noted $p, q, \ldots$ which are used to represent the value taken by an input variable (or feature). When referring to input variables/features we sometimes use the notation 'p' to distinguish it from the symbol p for atomic proposition. In this sense, the atomic proposition p should be read "the Boolean input variable 'p' takes value 1", while its negation $\neg p$ should be read "the Boolean input variable 'p' takes value 0".

We introduce a finite set Val to denote the *output values* (classifications, decisions) of the classifier. Elements of Val are also called *classes* in the jargon of classifiers. For this reason, we note them $c, c', \ldots$ For any $c \in Val$, we call $\mathtt{t}(c)$ a decision atom, to be read as "the actual decision (or output) takes value c", and have $Dec = \{\mathtt{t}(c) : c \in Val\}$. Finally, let $Atm = Atm_0 \cup Dec$ be the set of atomic formulas. Notice symbols c and p have different statuses: p is an atomic proposition representing an atomic fact, while c is not. This explains why c (an output value) and $\mathtt{t}(c)$ (an atomic formula representing the fact that the actual output has a certain value) are distinguished.

The modal language $\mathcal{L}(Atm)$ is hence defined by the following grammar:

[7] More recently, similar operators have been used in the context of the logic of functional dependence by Baltag & van Benthem [3].

$$\varphi \quad ::= \quad p \mid \mathtt{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi,$$

where p ranges over Atm_0, c ranges over Val, and X is a finite subset of Atm_0 which we note $X \subseteq^{\text{fin}} Atm_0$.

The set of atomic formulas occurring in a formula φ is noted $Atm(\varphi)$.

The formula $[X]\varphi$ has to be read "φ is necessary all features in X being equal" or "φ is necessary regardless of the truth or falsity of the atoms in $Atm_0 \setminus X$". Operator $\langle X\rangle$ is the dual of $[X]$ and is defined as usual: $\langle X\rangle\varphi =_{def} \neg[X]\neg\varphi$.

The language $\mathcal{L}(Atm)$ is interpreted relative to classifier models whose class is defined as follows.

Definition 1 (Classifier model)**.** *A classifier model (CM) is a tuple $C = (S, f)$ where:*

- $S \subseteq 2^{Atm_0}$ *is a set of states or input instances, and*
- $f : S \longrightarrow Val$ *is a decision (or classification) function.*

The class of classifier models is noted **CM**.

A pointed classifier model is a pair (C, s) with $C = (S, f)$ a classifier model and $s \in S$. Formulas in $\mathcal{L}(Atm)$ are interpreted relative to a pointed classifier model, as follows.

Definition 2 (Satisfaction relation)**.** *Let (C, s) be a pointed classifier model with $C = (S, f)$ and $s \in S$. Then:*

$$\begin{array}{rcl}
(C,s) \models p & \iff & p \in s, \\
(C,s) \models \mathtt{t}(c) & \iff & f(s) = c, \\
(C,s) \models \neg\varphi & \iff & (C,s) \not\models \varphi, \\
(C,s) \models \varphi \wedge \psi & \iff & (C,s) \models \varphi \textit{ and } (C,s) \models \psi, \\
(C,s) \models [X]\varphi & \iff & \forall s' \in S, \textit{ if } (s \cap X) = (s' \cap X) \textit{then } (C,s') \models \varphi.
\end{array}$$

We can think of a pointed model (C, s) as a pair (s, c) of f with $f(s) = c$. Thus, c is the output of the input instance s according to f. The condition $(s \cap X) = (s' \cap X)$, which induces an equivalence relation modulo X, indeed stipulates that s and s' are indistinguishable regarding the atoms (the features) in X. The formula $[X]\varphi$ is true at a state s if φ is true at all states that are modulo-X equivalent to state s. It has the *selectis paribus* (SP) (selected things being equal) interpretation "features in X being equal, necessarily φ holds (under possible perturbation on the other features)". When Atm_0 is finite, $[Atm_0 \setminus X]\varphi$ has the standard *ceteris paribus* (CP) interpretation "features other than X being equal, necessarily φ holds (under possible perturbation of the features in X)".[8] When $X = \emptyset$, $[\emptyset]$ is the S5 universal modality since every state is modulo-$\emptyset$ equivalent to all states.

[8] We thank Giovanni Sartor for drawing the distinction between CP and SP.

A formula φ of $\mathcal{L}(Atm)$ is said to be satisfiable relative to the class **CM** if there exists a pointed classifier model (C, s) with $C \in \mathbf{CM}$ such that $(C, s) \models \varphi$. It is said to be valid relative to **CM**, noted $\models_{\mathbf{CM}} \varphi$, if $\neg\varphi$ is not satisfiable relative to **CM**. Moreover, we say that that φ is valid in the classifier model $C = (S, f)$, noted $C \models \varphi$, if $(C, s) \models \varphi$ for every $s \in S$.

It is worth noting that every modality $[X]$ can be defined by means of the universal modality $[\emptyset]$. To show this, let us introduce the following abbreviation for every $Y \subseteq X \subseteq^{\text{fin}} Atm_0$:

$$\mathsf{cn}_{Y,X} =_{def} \bigwedge_{p \in Y} p \wedge \bigwedge_{p \in X \setminus Y} \neg p.$$

$\mathsf{cn}_{Y,X}$ can be seen as the syntactic expression of a valuation on X, and therefore represents a set of states in a classifier model satisfying the valuation. We have the following validity for the class **CM**:

$$\models_{\mathbf{CM}} [X]\varphi \leftrightarrow \Big(\bigwedge_{Y \subseteq X} \big(\mathsf{cn}_{Y,X} \rightarrow [\emptyset](\mathsf{cn}_{Y,X} \rightarrow \varphi)\big)\Big).$$

It means that $[X]\varphi$ is true at state s, if and only if, for whatever $Y \subseteq X$, if $s \cap X = Y$ then for any state s' such that $s' \cap X = Y$, φ is true at s'.

Let us close this section by formally introducing our running example.

Example 2 (Applicant Alice, formal)**.** *Let* $Atm = \{male, center, employed, owner\} \cup \{\mathsf{t}(1), \mathsf{t}(0)\}$*, where* 1 *and* 0 *stand for accepted and rejected respectively. Suppose* $C = (S, f)$ *is a CM such that* $S = 2^{Atm_0}$ *and*

$$C \models \big(\mathsf{t}(1) \leftrightarrow ((male \wedge employed) \vee (employed \wedge owner))\big).$$

Consider the state $s = \{center, employed\}$*. Then,* s *stands for the instance Alice and* f *for the classifier in Example 1 such that* $f(s) = 0$*.*

Now Alice is asking for explanations of the decision/classification, e.g., 1) which of her features (necessarily) lead to the current decision, 2) changing which features would make a difference, 3) perhaps most importantly, whether the decision for her is biased. In Section 5 we will show how to use the language $\mathcal{L}(Atm)$ and its semantics to answer these questions.

2.2 Discussion

In this subsection we discuss in more detail some subtleties of classifier models in relation with the modal language $\mathcal{L}(Atm)$ which is interpreted over them.

X-Completeness In the definition of classifier model (Definition 1) given above, we stipulated that the set of states S does not necessarily include all possible input instances of a classifier. More generally, according to our definition, a classifier model could be incomplete with respect to a set of atoms X from Atm_0, that is, there could be a truth assignment for the atoms in X which

is not represented in the model. Incompleteness of a classifier model is justified by the fact that in certain domains of application hard constraints exist which prevent for some input instance to occur. For example, a hard constraint may impose that a male cannot be pregnant (i.e., all states in which atoms *male* and *pregnant* are true should be excluded from the model).

However, it is interesting to see how completeness of a classifier with respect to a finite set of features can be represented in our semantics. This is what the following definition specifies.

Definition 3 (X-completeness)**.** *Let $C = (S, f)$ be a classifier model and $X \subseteq^{\text{fin}} Atm_0$. Then, C is said to be X-complete, if $\forall X' \subseteq X, \exists s \in S$ such that $s \cap X = X'$.*

In plain words, the definition means that any truth assignment for the atoms in X is represented by some state of the model. As the following proposition indicates, the class of X-complete CMs can be syntactically represented. The proof is straightforward and omitted.

Proposition 1. *Let $C = (S, f)$ be a CM and $X \subseteq^{\text{fin}} Atm_0$. C is X-complete if and only if $\forall s \in S$, we have $(C, s) \models \mathsf{Comp}(X)$, with*

$$\mathsf{Comp}(X) =_{def} \bigwedge_{X' \subseteq X} \langle\emptyset\rangle \mathsf{cn}_{X',X}.$$

X-Definiteness In certain situations, there might be a portion of the feature space which is irrelevant for the classifier's decision. For example, in the Alice's example the fact of renting an apartment in the city center (the feature *center*) plays no role in the classification. In this case, we say that the classifier is definite with respect to the subset of features $\{male, employed, owner\}$.

More generally, a classifier is said to be definite with respect to a set of features X if its decision is only determined by the variables in X, that is to say, the variables in the complementary set $Atm_0 \setminus X$ play no role in the classifier's decision. In other words, the classifier is said to be X-definite if its decision is independent of the variables in $Atm_0 \setminus X$.

The following definition introduces the concept of X-definiteness formally.

Definition 4 (X-definiteness)**.** *Let $C = (S, f)$ be a classifier model and $X \subseteq^{\text{fin}} Atm_0$. Then, C is said to be X-definite, if $\forall s, s' \in S$, if $s \cap X = s' \cap X$ then $f(s) = f(s')$.*

X-definiteness is tightly related to the notion of dependence studied in (propositional) dependence logic [53]. The latter focuses on so-called dependence atoms of the form $=(X, q)$ where q is a propositional variable and X is a finite set of propositional variables. The latter expresses the fact that the truth value of the propositional variable q only depends on the truth values of the propositional variables in X. It turns out that dependence atoms can be expressed in our *ceteris paribus* modal language $\mathcal{L}(Atm)$ as abbreviations:

$$=(X, q) =_{def} [\emptyset]\big((q \to [X]q) \wedge (\neg q \to [X]\neg q)\big).$$

Interestingly, the notion of X-definiteness is expressible in our modal language by means of the dependence atoms. This is what the following proposition indicates.

Proposition 2. *Let $C = (S, f)$ be a CM and $X \subseteq^{\text{fin}} Atm_0$. C is X-definite if and only if $\forall s \in S, (C, s) \models \mathsf{Defin}(X)$ with*

$$\mathsf{Defin}(X) =_{def} \bigwedge_{c \in Val} =\big(X, \mathsf{t}(c)\big).$$

We conclude this section by mentioning some remarkable properties of X-definiteness. The first fact to be noticed is that X-definiteness is upward closed.

Fact 1. *For every $C \in \mathbf{CM}$ and $X \subseteq Y \subseteq^{\text{fin}} Atm_0$, if C is X-definite then C is Y-definite too.*

Secondly, X-definiteness for some $X \subseteq^{\text{fin}} Atm_0$ is guaranteed in the finite-variable case.

Fact 2. *For every $C \in \mathbf{CM}$, if Atm_0 is finite then C is Atm_0-definite.*

This does not hold in the infinite case.

Fact 3. *If Atm_0 is countably infinite and $|Val| > 1$ then there exists $C \in \mathbf{CM}$ such that, for all $X \subseteq^{\text{fin}} Atm_0$, C is not X-definite.*

The previous fact is witnessed by any CM $C = (S, f)$ such that

- $S = 2^{Atm_0}$,
- $f(Atm_0) = 1$,
- $\forall s \in S$, if $|Atm_0 \triangle s| = 1$ then $f(s) = 0$,

where $Dec = \{0, 1\}$ and $\triangle$ denotes symmetric difference, viz., $s \triangle s' = (s \setminus s') \cup (s' \setminus s)$. It is easy to show that a CM so defined is not X-definite for any $X \subseteq^{\text{fin}} Atm_0$.

3 Axiomatization and Complexity

In this section, we provide axiomatics for our logical setting. We distinguish the finite-variable from the infinite-variable case. We moreover prove complexity results for satisfiability checking for both cases. But before, we will first introduce an alternative Kripke semantics for the interpretation of the language $\mathcal{L}(Atm)$. It will allow us to use the standard canonical model technique for proving completeness. Indeed, this technique cannot be directly applied to CMs in the infinite-variable case since our modal language is not expressive enough to capture the "functionality" property of CMs when Atm_0 is infinite. We think it would be possible to apply the canonical model argument directly to CMs in the finite-variable case. But we leave this to future work.

3.1 Alternative Kripke Semantics

In our alternative semantics the concept of classifier model is replaced by the following concept of decision model. It is a multi-relational Kripke structure with one accessibility relation per finite set of atoms *plus* a number of constraints over the accessibility relations and the valuation function for the atomic propositions.

Definition 5 (Decision model)**.** *A decision model (DM) is a tuple* $M = \big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ *such that* W *is a set of possible worlds,* $V : W \longrightarrow 2^{Atm}$ *is a valuation function for atomic formulas, and* $\forall w, v \in W, c, c' \in Val$ *the following constraints are satisfied:*

(C1) $w \equiv_X v$ *iff* $V_X(w) = V_X(v)$,

(C2) $V_{Dec}(w) \neq \emptyset$,

(C3) *if* $\mathsf{t}(c), \mathsf{t}(c') \in V(w)$ *then* $c = c'$,

(C4) *if* $V_{Atm_0}(w) = V_{Atm_0}(v)$ *then* $V_{Dec}(w) = V_{Dec}(v)$;

where $V_X(w)$ *abbreviates* $V(w) \cap X$*. The class of DMs is noted* **DM**.

A DM $\big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ is called finite if W is finite. The class of finite-DM is noted **finite-DM**.

The interpretation of formulas in $\mathcal{L}(Atm)$ relative to a pointed DM goes as follows.

Definition 6 (Satisfaction relation)**.** *Let* $\big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ *be a DM and let* $w \in W$*. Then,*

$$\begin{array}{rcl}
(M, w) \models p & \iff & p \in V(w), \\
(M, w) \models \mathsf{t}(c) & \iff & \mathsf{t}(c) \in V(w), \\
(M, w) \models \neg\varphi & \iff & (M, w) \not\models \varphi, \\
(M, w) \models \varphi \wedge \psi & \iff & (M, w) \models \varphi \text{ and } (M, w) \models \psi, \\
(M, w) \models [X]\varphi & \iff & \forall v \in W, \text{ if } w \equiv_X v \text{ then } v \models \varphi.
\end{array}$$

Validity and satisfiability of formulas in $\mathcal{L}(Atm)$ relative to class **DM** (resp. **finite-DM**) is defined in the usual way.

The following theorem appears obvious, since it only has to do with the matter whether the decision function (classifier) f is given as a constituent of the model or induced from the model. Notice that it holds regardless of Atm_0 being finite or countably infinite.

Theorem 1. *Let* $\varphi \in \mathcal{L}(Atm)$*. Then,* φ *is satisfiable relative to the class* **CM** *if and only if it is satisfiable relative to the class* **DM**.

3.2 Axiomatization: Finite-Variable Case

In this section we provide a sound and complete axiomatics for the language $\mathcal{L}(Atm)$ relative to the formal semantics defined above under the assumption that the set of atomic propositions Atm_0 is finite.

Definition 7 (Logic BCL)**.** *We define* BCL *(Binary Classifier Logic) to be the extension of classical propositional logic given by the following axioms and rules of inference:*

$$\big([\emptyset]\varphi \wedge [\emptyset](\varphi \to \psi)\big) \to [\emptyset]\psi \qquad (\mathbf{K}_{[\emptyset]})$$

$$[\emptyset]\varphi \to \varphi \qquad (\mathbf{T}_{[\emptyset]})$$

$$[\emptyset]\varphi \to [\emptyset][\emptyset]\varphi \qquad (\mathbf{4}_{[\emptyset]})$$

$$\varphi \to [\emptyset]\langle\emptyset\rangle\varphi \qquad (\mathbf{B}_{[\emptyset]})$$

$$[X]\varphi \leftrightarrow \bigwedge_{Y \subseteq X} \big(\mathsf{cn}_{Y,X} \to [\emptyset](\mathsf{cn}_{Y,X} \to \varphi)\big) \qquad (\mathbf{Red}_{[\emptyset]})$$

$$\bigvee_{c \in Val} \mathsf{t}(c) \qquad (\mathbf{AtLeast})$$

$$\mathsf{t}(c) \to \neg\mathsf{t}(c') \text{ if } c \neq c' \qquad (\mathbf{AtMost})$$

$$\bigwedge_{Y \subseteq^{\text{fin}} Atm_0} \Big(\big(\mathsf{cn}_{Y,Atm_0} \wedge \mathsf{t}(c)\big) \to [\emptyset]\big(\mathsf{cn}_{Y,Atm_0} \to \mathsf{t}(c)\big)\Big) \qquad (\mathbf{Funct})$$

$$\frac{\varphi \to \psi, \quad \varphi}{\psi} \qquad (\mathbf{MP})$$

$$\frac{\varphi}{[\emptyset]\varphi} \qquad (\mathbf{Nec}_{[\emptyset]})$$

It can be seen that $[\emptyset]$ is an S5 style modal operator, $\mathbf{Red}_{[\emptyset]}$ reduces any $[X]$ to $[\emptyset]$. **AtLeast**, **AtMost**, **Funct** represent the decision function syntactically and that every expression cn_{Y,Atm_0} maps to some unique $\mathsf{t}(c)$.

A decision model can contain two copies of the same input instance, while a classifier model cannot. Thus, Theorem 1 above shows that our modal language is not powerful enough to capture this difference between CMs and DMs. Axiom **Funct** intervenes in the finite-variable case to guarantee that two copies of the same input instance (that may exist in a DM) have the same output value. The expression cn_{Y,Atm_0} used in the axiom is an instance of the abbreviation we defined in Section 2.1. It represents a specific input instance. Notice that this abbreviation is only legal when Atm_0 is finite. Otherwise it would be the abbreviation of an infinite conjunction which is not allowed, since our modal language is finitary.

The proof of the following theorem is entirely standard and based on a canonical model argument.

Theorem 2. *Let Atm_0 be finite. Then, the logic* BCL *is sound and complete relative to the class* **DM**.

The main result of this subsection is now a corollary of Theorems 1 and 2.

Corollary 1. *Let Atm_0 be finite. Then, the logic* BCL *is sound and complete relative to the class* **CM**.

3.3 Axiomatization: Infinite-Variable Case

In Section 3.2, we have assumed that the set of atomic propositions Atm_0 representing input features is finite. In this section, we drop this assumption and prove completeness of the resulting logic.

An essential feature of the logic BCL is the "functionality" Axiom **Funct**. Such an axiom cannot be represented in a finitary way when assuming that the set Atm_0 is countably infinite. For this reason, it has to be dismissed and the logic weakened.

Definition 8 (Logic WBCL)**.** *The logic* WBCL *(Weak* BCL*) is defined by all principles of logic* BCL *given in Definition 7 except Axiom* ***Funct****.*

In order to obtain the completeness of WBCL relative to the class **CM**, besides decision models (DMs), we need additionally quasi-decision models (QDMs).

Definition 9 (Quasi-DM)**.** *A quasi-DM is a tuple $M = \big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ where W, $(\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}$ and V are defined as in Definition 5 and which satisfies all constraints of Definition 5 except* **C4**. *The class of quasi-DMs is noted* ***QDM****.*

A quasi-DM $\big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ is said to be finite if W is finite. The class of finite quasi-DMs is noted **finite-QDM**.

Semantic interpretation of formulas in $\mathcal{L}(Atm)$ relative to quasi-DMs is analogous to semantic interpretation relative to DMs given in Definition 6. Moreover, validity and satisfiability of formulas in $\mathcal{L}(Atm)$ relative to class **QDM** (resp. **finite-QDM**) is again defined in the usual way.

We are going to show the equivalence between **QDM** and **CM** step by step. The following theorem is proven by filtration.

Theorem 3. *Let Atm_0 be countably infinite and $\varphi \in \mathcal{L}(Atm)$. Then, φ is satisfiable relative to the class* ***QDM*** *if and only if φ is satisfiable relative to the class* ***finite-QDM****.*

Then, let us establish the crucial fact that, in the infinite-variable case, the language $\mathcal{L}(Atm)$ cannot distinguish finite-DMs from finite-QDMs. We are going to prove that any formula φ satisfiable in a finite-QDM M is also satisfiable in some finite-DM M'. Since the only condition to worry is **C4**, we just need to transform the valuation function of M to guarantee that **C4** holds while still satisfying φ.

Theorem 4. *Let Atm_0 be countably infinite and $\varphi \in \mathcal{L}(Atm)$. Then, φ is satisfiable relative to the class* ***finite-QDM*** *if and only if φ is satisfiable relative to the class* ***finite-DM****.*

Recall Theorem 1 shows that $\mathcal{L}(Atm)$ can not distinguish between CMs and DMs regardless of Atm_0 being finite or infinite. Thus, we obtain the desired equivalence between model classes **QDM** and **CM** in the infinite-variable case, as a corollary of Theorems 1, 3 and 4. This fact is highlighted by Figure 1. More generally, Figure 1 shows that when Atm_0 is countably infinite the five semantics for the modal language $\mathcal{L}(Atm)$ are all equivalent, since from every node in the graph we can reach all other nodes.

Theorem 5. *Let Atm_0 be countably infinite and $\varphi \in \mathcal{L}(Atm)$. Then, φ is satisfiable relative to the class* **QDM** *if and only if φ is satisfiable relative to the class* **CM**.

As a consequence, we are in position of proving that the logic WBCL is also sound and complete for the corresponding classifier model semantics, under the infinite-variable assumption. The only missing block is the following completeness theorem. The proof is similar to the proof of Theorem 2 (with the only difference that the canonical QDM does not need to satisfy **C4**), and omitted.

Theorem 6. *Let Atm_0 be countably infinite. Then, the logic* WBCL *is sound and complete relative to the class* **QDM**.

The main result of this subsection turns out to be a direct corollary of Theorems 5 and 6.

Corollary 2. *Let Atm_0 be countably infinite. Then, the logic* WBCL *is sound and complete relative to the class* **CM**.

3.4 Complexity Results

Let us now move from axiomatics to complexity issues. Our first result is about complexity of checking satisfiability for formulas in $\mathcal{L}(Atm)$ relative to the class **CM** when Atm_0 is finite and fixed. It is in line with the satisfiability checking problem of the modal logic S5 which is known to be polynomial in the finite-variable case [18].

Theorem 7. *Let Atm_0 be finite and fixed. Then, checking satisfiability of formulas in $\mathcal{L}(Atm)$ relative to* **CM** *can be done in polynomial time.*

As the following theorem indicates, the satisfiability checking problem becomes intractable when dropping the finite-variable assumption.

Theorem 8. *Let Atm_0 be countably infinite. Then, checking satisfiability of formulas in $\mathcal{L}(Atm)$ relative to* **CM** *is NEXPTIME-complete.*

Let us consider the following fragment $\mathcal{L}^{\{[\emptyset]\}}(Atm)$ of the language $\mathcal{L}(Atm)$ where only the universal modality $[\emptyset]$ is allowed:

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\emptyset]\varphi.$$

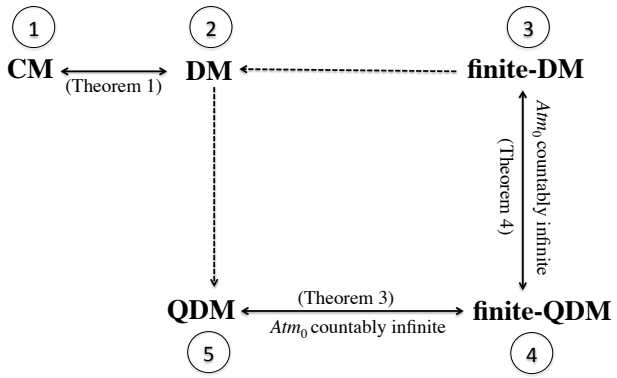

Figure 1: Relations between semantics for the modal language $\mathcal{L}(Atm)$. An arrow means that satisfiability relative to the first class of structures implies satisfiability relative to the second class of structures. Full arrows correspond to the results stated in Theorems 1, 3 and 4. Dotted arrows denote relations that follow straightforwardly given the inclusion between classes of structures. The bidirectional arrows connecting node 3 with node 4 and node 4 with node 5 only apply to the infinite-variable case.

Clearly, satisfiability checking for formulas in $\mathcal{L}^{\{[\emptyset]\}}(Atm)$ remains polynomial when there are only finitely many primitive propositions. As the following theorem indicates, complexity decreases from NEXPTIME to NP when restricting to the fragment $\mathcal{L}^{\{[\emptyset]\}}(Atm)$ under the infinite-variable assumption.

Theorem 9. *Let Atm_0 be countably infinite. Then, checking satisfiability of formulas in $\mathcal{L}^{\{[\emptyset]\}}(Atm)$ relative to* **CM** *is NP-complete.*

The complexity results of this section are summarized in Table 1.

	Fixed finite variables	**Infinite variables**
Fragment $\mathcal{L}^{\{[\emptyset]\}}(Atm)$	Polynomial	NP-complete
Full language $\mathcal{L}(Atm)$	Polynomial	NEXPTIME-complete

Table 1: Summary of complexity results

4 Counterfactual Conditional

In this section we investigate a simple notion of counterfactual conditional for binary classifiers, inspired from Lewis' notion [27]. In Section 5, we will elucidate its connection with the notion of explanation.

We start our analysis by defining the following notion of similarity between states in a classifier model relative to a finite set of features X.

Definition 10 (Similarity between states)**.** *Let $C = (S, f)$ be a classifier model, $s, s' \in S$ and $X \subseteq^{\text{fin}} Atm_0$. The similarity between s and s' in S relative to the set of features X, noted $sim_C(s,s',X)$, is defined as follows:*

$$sim_C(s,s',X) = |\{p \in X : (C,s) \models p \text{ iff } (C,s') \models p\}|.$$

A dual notion of distance between worlds can defined from the previous notion of similarity:

$$dist_C(s,s',X) = |X| - sim_C(s,s',X).$$

This notion of distance is in accordance with [11] in knowledge revision.[9]

The following definition introduces the notion of counterfactual conditional as an abbreviation. It is a form of relativized conditional, i.e., a conditional with respect to a finite set of features.[10]

Definition 11 (Counterfactual conditional)**.** *We write $\varphi \Rightarrow_X \psi$ to mean that "if φ was true then ψ would be true, relative to the set of features X" and define it as follows:*

$$\varphi \Rightarrow_X \psi =_{def} \bigwedge_{0 \leq k \leq |X|} \big(\mathsf{maxSim}(\varphi,X,k) \rightarrow \bigwedge_{Y \subseteq X : |Y| = k} [Y](\varphi \rightarrow \psi)\big),$$

with

$$\mathsf{maxSim}(\varphi,X,k) =_{def} \bigvee_{Y \subseteq X : |Y| = k} \langle Y \rangle \varphi \wedge \bigwedge_{Y \subseteq X : k < |Y|} [Y] \neg \varphi.$$

As the following proposition highlights, the previous definition of counterfactual conditional is in line with Lewis' view: the conditional holds if all closest worlds to the actual world in which the antecedent is true satisfy the consequent of the conditional.

Proposition 3. *Let $C = (S, f)$ be a classifier model, $s \in S$ and $X \subseteq^{\text{fin}} Atm_0$. Then, $(C, s) \models \varphi \Rightarrow_X \psi$ if and only if $closest_C(s,\varphi,X) \subseteq ||\psi||_C$, where*

$$closest_C(s,\varphi,X) = \underset{s' \in ||\varphi||_C}{\arg\max}\ sim_C(s,s',X),$$

and for every $\varphi \in \mathcal{L}(Atm)$:

$$||\varphi||_C = \{s \in S : (C,s) \models \varphi\}.$$

[9] There are other options besides measuring distance by cardinality, e.g., distance in sense of subset relation as [6]. We will consider them in further research.

[10] A similar approach to conditional is presented in [15]. They also refine Lewis' semantics for counterfactuals by selecting the closest worlds according to not only the actual world and antecedent, but also a set of formulas noted Γ. The main technical difference is that they allow any counterfactual-free formula as a member of Γ, while in our setting X only contains atomic formulas.

For notational convenience, we simply write $\varphi \Rightarrow \psi$ instead of $\varphi \Rightarrow_{Atm_0} \psi$, when Atm_0 is finite. Formula $\varphi \Rightarrow \psi$ captures the standard notion of conditional of conditional logic. One can show that $\Rightarrow$ satisfies all semantic conditions of Lewis' logic VC.[11] However, when Atm_0 is infinite, $\varphi \Rightarrow \psi$ is not a well-formed formula since it ranges over an infinite set of atoms. In that case $\varphi \Rightarrow_X \psi$ has to be always indexed by some finite X.

The interesting aspect of the previous notion of counterfactual conditional is that it can be used to represent a binary classifier's approximate decision for a given instance. Let us suppose the set of decision values *Val* includes a special symbol ? meaning that the classifier has no sufficient information enabling it to classify an instance in a precise way. More compactly, ? *is interpreted as* that the classifier abstains from making a precise decision. In this situation, the classifier can try to make an approximate decision: it considers the closest instances to the actual instance for which it has sufficient information to make a decision and checks whether the decision is uniform among all such instances. In other words, c is the classifier's approximate classification of (or decision for) the actual instance relative to the set of features X, noted $\mathsf{apprDec}(X,c)$, if and only if "if a precise decision was made relative to the set of features X, then this decision would be c". Formally:

$$\mathsf{apprDec}(X,c) =_{def} \big(\bigvee_{c' \in Val : c' \neq ?} \mathsf{t}(c') \big) \Rightarrow_X \mathsf{t}(c).$$

The following proposition provides two interesting validities.

Proposition 4. *Let Atm_0 be finite, $c, c' \in Val \setminus \{?\}$. Then,*

$$\models_{\mathbf{CM}} \mathsf{apprDec}(X,c) \rightarrow \neg \mathsf{apprDec}(X,c') \text{ if } c \neq c',$$
$$\models_{\mathbf{CM}} \mathsf{t}(c) \rightarrow \mathsf{apprDec}(Atm_0, c).$$

According to the first validity, a classifier cannot make two different approximate decisions relative to a fixed set of features X.

According to the second validity, if the classifier is able to make a precise decision for a given instance, then its approximate decision coincides with it. This second validity works since the actual state/instance is the only closest state/instance to itself. Therefore, it the actual state/instance has a precise classification c, all its closest states/instances also have it.

It is worth noting that the following formula is not valid relative to the class **CM**:

$$\bigvee_{c \in Val \setminus \{?\}} \mathsf{apprDec}(X,c).$$

This means that a classifier may be unable to approximately classify the actual instance. The reason is that there could be different closest states to the actual one with different classifications.

[11] A remarkable fact is that not all $\Rightarrow_X$ satisfy the *strong centering* condition, which says that the actual world is the only closest world when the antecedent is already true there. To see it, consider a toy classifier model (C, s) such that $S = \{s, s', s'', s'''\}$ with $s = \{p, q\}$, $s' = \{p\}$, $s'' = \{q\}$, $s''' = \emptyset$. We have $closest_C(s,p,\{p\}) = \{s, s'\}$, rather than $closest_C(s,p,\{p\}) = \{s\}$.

5 Explanations and Biases

In this section, we are going to formalize some existing notions of explanation of classifiers in our logic, and deepen the current study from a (finitely) Boolean setting to a multi-valued output, partial domain and possibly infinite-variable setting. For this purpose it is necessary to introduce the following notations.

Let λ denote a conjunction of finitely many literals, where a literal is an atom p or its negation $\neg p$. We write $\lambda \subseteq \lambda'$, call λ a part (subset) of λ', if all literals in λ also occur in λ'; and $\lambda \subset \lambda'$ if $\lambda \subseteq \lambda'$ but not $\lambda' \subseteq \lambda$. By convention $\top$ is a term of zero conjuncts. In particular, suppose λ is $\mathsf{cn}_{X,Y}$ for some $X \subseteq Y \subseteq^{\text{fin}} Atm_0$, then $\overline{\lambda}$ will denote the conjunction resulting from "flipping" (or "perturbing") all literals of λ, i.e., $\mathsf{cn}_{Y \setminus X,Y}$.

In the glossary of Boolean classifiers, λ is called a *term* or *property* (of an instance). The set of terms is noted *Term*. We use $Term(X)$ to denote all terms whose atoms are in X. Additionally, to define the notion of bias we distinguish the set of protected features $\mathsf{PF} \subseteq Atm_0$, like 'gender' and 'race', and the set of non-protected features $\mathsf{NF} = Atm_0 \setminus \mathsf{PF}$.

Notice that in this section the cardinality of Atm_0 matters. Notions and results in Section 5.1 (without special instruction) apply to both Atm_0 finite and Atm_0 countably infinite. On the contrary, in Sections 5.2 and 5.3, we restrict to the case Atm_0 finite, which is due to the use of formulas $[Atm_0 \setminus X]\varphi, [\mathsf{NF}]\varphi$ and $[\mathsf{PF}]\varphi$ there. We clarify it here instead of clarifying it below repeatedly.

5.1 Prime Implicant and Abductive Explanation

We are in position to formalize the notion of *prime implicant*, which plays a fundamental role in the theory of Boolean functions since [39].

Definition 12 (Prime implicant (PImp))**.** *We write* $\mathsf{PImp}(\lambda, c)$ *to mean that* λ *is a* prime implicant *for* c *and define it as follows:*

$$\mathsf{PImp}(\lambda, c) =_{def} [\emptyset]\Big(\lambda \to \big(\mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} \langle Atm(\lambda) \setminus \{p\}\rangle \neg \mathsf{t}(c)\big)\Big).$$

It is a proper extension of the definition of prime implicant in the Boolean setting since it is a term λ such that 1) it necessarily implies the actual classification (why it is called an *implicant*); 2) any of its proper subsets fails to necessarily imply the actual classification (why it is called *prime*). Notice that being a prime implicant is a global property of the classifier, though we formalize it by means of a pointed model. The syntactic abbreviation for prime implicant can be better understood by observing that for a given CM $C = (S, f)$ and $s \in S$, we have:

$$\begin{aligned}(C, s) \models \mathsf{PImp}(\lambda, c) \text{ iff } &(i)\ \forall s' \in S, \text{ if } (C, s') \models \lambda \text{ then } (C, s') \models \mathsf{t}(c); \text{ and} \\ &(ii)\ \forall \lambda' \subset \lambda, \exists s' \in S \text{ such that } (C, s') \models \lambda' \wedge \neg \mathsf{t}(c).\end{aligned}$$

To explain the actual classification of a given input, some XAI researchers consider prime implicants which are actually true. We use the terminology by [24] and call them abductive explanations (AXp).

Definition 13 (Abductive explanation (AXp))**.** *We write* $\mathsf{AXp}(\lambda, c)$ *to mean that* λ abductively explains *the decision* c *and define it as follows:*

$$\mathsf{AXp}(\lambda, c) =_{def} \lambda \wedge \mathsf{PImp}(\lambda, c).$$

AXp is a local explanation, because λ is not only a prime implicant for the classification, but also a property of the actual instance to be classified. AXp can be expanded to highlight its connection with the notion of variance/invariance.

Proposition 5. *Let* $\lambda \in Term$ *and* $c \in Val$*. Then, we have the following validity:*

$$\models_{\mathbf{CM}} \mathsf{AXp}(\lambda, c) \leftrightarrow \big(\lambda \wedge [Atm(\lambda)]\mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} \langle Atm(\lambda) \setminus \{p\}\rangle \neg \mathsf{t}(c)\big).$$

The formula $[Atm(\lambda)]\mathsf{t}(c)$ expresses the idea of invariance under intervention (perturbation): as long as the explanans variables are kept fixed, namely the variables in λ, any perturbation on the other variables does not change the explanandum, namely classification c.

Many names besides AXp are found in literature, e.g., *PI-explanation* [41] and *sufficient reason* [12]. Darwiche and Hirth in [12] proved that any decision has a sufficient reason in the Boolean setting. The result is not a surprise, for a Boolean function always has a prime implicant, since by definition the arity of a Boolean function is always finite. However, since we allow functions with infinitely many variables, AXps are not guaranteed to exist in general.

Fact 4. *Let* Atm_0 *be countably infinite and* $|Val| > 1$*. Then, there exists some* $C = (S, f)$*,* $s \in S$*, such that* $\exists c \in Val, \forall \lambda \in Term$*,* $(C, s) \models \neg\mathsf{AXp}(\lambda, c)$*.*

The statement can be proved by exhibiting the same infinite countermodel as in Fact 3 in Section 2.2. However, if a CM is X-definite for some $X \subseteq^{\text{fin}} Atm_0$, then every state has an AXp, even when the CM is infinite.

Proposition 6. *Let* $C = (S, f) \in \mathbf{CM}$ *and* $X \subseteq^{\text{fin}} Atm_0$*. If* C *is* X*-definite then* $\forall s \in S, \exists \lambda \in Term$ *such that* $(C, s) \models \mathsf{AXp}\big(\lambda, f(s)\big)$*.*

Lastly, let us continue with the Alice example.

Example 3. *Recall the state of Alice* $s = \{center, employed\}$*. We have* $(C, s) \models \mathsf{AXp}(\neg male \wedge \neg owner, 0)$*, namely that Alice's being female and not owning a property abductively explains the rejection of her application.*

5.2 Contrastive Explanation (CXp)

AXp is a minimal part of the actual instance guaranteeing the current decision. A natural counterpart of AXp is contrastive explanation (CXp, named in [23]).

Definition 14 (Contrastive explanation (CXp)). *We write* $\mathsf{CXp}(\lambda, c)$ *to mean that* λ contrastively explains *the decision* c *and define it as follows:*

$$\mathsf{CXp}(\lambda, c) =_{def} \lambda \wedge \langle Atm_0 \setminus Atm(\lambda)\rangle \neg \mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} [(Atm_0 \setminus Atm(\lambda)) \cup \{p\}]\mathsf{t}(c).$$

The definition says nothing but 1) λ is part of the actual input instance; 2) if the values of all variables in λ are changed while the values of the other variables are kept fixed, then the actual classification may change; 3) the classification will not change, if the variables outside λ and at least one variable in λ keep their actual values. The latter captures a form of necessity: when the values of the variables outside λ are kept fixed, all variables in λ should be *necessarily* perturbed to change the actual classification.

The syntactic abbreviation for contrastive explanation can be better understood by observing that for a given CM $C = (S, f)$ and $s \in S$, we have:

$(C, s) \models \mathsf{CXp}(\lambda, c)$ iff (i) $(C, s) \models \lambda$;
(ii) $\exists s' \in S$ s.t. $s \triangle s' = Atm(\lambda)$ and $(C, s') \models \neg \mathsf{t}(c)$; and
(iii) $\forall s' \in S$, if $s \triangle s' \subset Atm(\lambda)$ then $(C, s') \models \mathsf{t}(c)$.

CXp has a counterfactual flavor since it answers to question: would the classification differ from the actual one, if the values of all variables in the explanans were different? So, there seems to be a connection with the notion of counterfactual conditional we introduced in Section 4. Actually in XAI, many researchers consider contrastive explanation and counterfactual explanation either closely related [48] or even interchangeable [42]. The following proposition sheds light on this point.

Proposition 7. *Let* λ *be a term and let* l *be a literal. Then, we have the following two validities:*

$$\models_{\mathbf{CM}} \mathsf{CXp}(\lambda, c) \rightarrow \Big(\mathsf{t}(c) \wedge \big(\overline{\lambda} \Rightarrow \neg \mathsf{t}(c)\big)\Big),$$
$$\models_{\mathbf{CM}} \mathsf{CXp}(l, c) \leftrightarrow \Big(\mathsf{t}(c) \wedge \big(\neg l \Rightarrow \neg \mathsf{t}(c)\big)\Big).$$

According to the first validity, in the general case contrastive explanation implies counterfactual explanation. According to the second validity, when the explanans is a literal (a single-conjunct term), contrastive explanation coincides with counterfactual explanation. In particular, literal l contrastively explains the decision c if and only if (i) the actual decision is c and (ii) if literal l was perturbed, then the decision would be different from c. In other words, in the "atomic" case CXp is the same as counterfactual explanation.

Note that the right-to-left direction of the first validity does not necessarily hold. To see this, it is sufficient to suppose that $Atm_0 = \{p, q\}$ and $Dec = \{0, 1\}$ and to consider the CM (S, f) such that $S = 2^{Atm_0}$ with $f\big(\{p, q\}\big) = 0$ and

$f(\{p\}) = f(\{q\}) = f(\emptyset) = 1$. It is easy to check that in the model so defined we have

$$(C, \{p, q\}) \models \mathsf{t}(0) \wedge (\overline{p \wedge q} \Rightarrow \neg \mathsf{t}(0)),$$

but at the same time,

$$(C, \{p, q\}) \models \neg \mathsf{CXp}(p \wedge q, 0).$$

The problem is that the model fails to satisfy the necessity condition of contrastive explanation: it is not necessary to perturb both literals in $p \wedge q$ to change the actual decision from 0 to 1, it is sufficient to perturb one of them. We can conclude that CXp is a special kind of counterfactual explanation with the additional requirement of necessity for the explanans.

Example 4. *In Alice's case, we have* $(C, s) \models \mathsf{CXp}(\neg male, 0) \wedge \mathsf{CXp}(\neg owner, 0)$. *This means that both Alice's being female and not owning property contrastively explain the rejection. Moreover, we have* $(C, s) \models (\neg male \vee \neg owner) \Rightarrow \mathsf{t}(1)$, *namely if Alice was a male or an owner (of an immobile property), then her application would have been accepted.*

Moreover, since the feature 'gender' is hard to change, owing a property is the (relatively) *actionable* explanation for Alice,[12] if she intends to comply with the classifier's decision. But surely Alice has another option, i.e., alleging the classifier as biased. As we will see in the next subsection, an application of CXp is to detect decision biases in a classifier.

5.3 Decision Bias

A primary goal of XAI is to detect and avoid biases. Bias is understood as making decision with respect to some protected features, e.g., 'race', 'gender' and 'age'.

There is a widely accepted notion of decision bias in the setting of Boolean functions which can be represented in our Example 2 (see [12, 22]). Intuitively, the rejection for Alice is biased if there is another applicant, say Bob, who only differs from Alice on the protected feature 'gender', but gets accepted.

Definition 15 (Decision bias)**.** *We write* $\mathsf{Bias}(c)$ *to mean that the decision* c is biased *and define it as follows:*

$$\mathsf{Bias}(c) =_{def} \mathsf{t}(c) \wedge \langle \mathsf{NF} \rangle \neg \mathsf{t}(c).$$

The definition says that the decision c is biased at a given state s, if (i) $f(s) = c$, and (ii) $\exists s' \in S$ such that $s \triangle s' \subseteq \mathsf{PF}$ and $f(s') \neq c$. The latter, in plain words, requires another instance s', which only differs from s on some protected features, but obtains a different classification.

As we stated, CXp can be used to detect decision biases. The following result makes the statement precise.

[12] For the significance of actionability in XAI, see e.g. [42].

Proposition 8. *We have the following validity:*

$$\models_{\mathbf{CM}} \mathsf{Bias}(c) \leftrightarrow \bigvee_{Atm(\lambda)\subseteq \mathsf{PF}} \mathsf{CXp}(\lambda, c).$$

Let us end up the whole section by answering the last question regarding Alice raised at the end of Section 2.1.

Example 5. *Split Atm_0 in Example 2 into $\mathsf{PF} = \{male, center\}$ and $\mathsf{NF} = \{employed, owner\}$. We then have $(C, s) \models \mathsf{Bias}(0) \wedge \mathsf{CXp}(male, 0) \wedge \big(\neg male \Rightarrow \mathsf{t}(1)\big)$. The decision for Alice is biased since 'gender' is the protected feature which contrastively explains the rejection, and if Alice was male, her application would have been accepted.*

6 Extensions

In this section, we briefly discuss two interesting extensions of our logical framework and analysis of binary classifiers. Their full development is left for future work.

6.1 Dynamic Extension

The first extension we want to discuss consists in adding to the language $\mathcal{L}(Atm)$ dynamic operators of the form $[c{:=}\varphi]$ with $c \in Val$, where $c{:=}\varphi$ is a kind of assignment in the sense of [43, 47] and the formula $[c := \varphi]\psi$ has to be read "ψ holds after every decision is set to c in context φ". The resulting language, noted $\mathcal{L}^{dyn}(Atm)$, is defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi \mid [c{:=}\varphi]\psi,$$

where p ranges over Atm_0, c ranges over Val, and $X \subseteq^{\text{fin}} Atm_0$. The interpretation of formula $[c{:=}\varphi]\psi$ relative to a pointed classifier model (C, s) with $C = (S, f)$ goes as follows:

$$(C, s) \models [c{:=}\varphi]\psi \iff (C^{c:=\varphi}, s) \models \psi,$$

where $C^{c:=\varphi} = (S, f^{c:=\varphi})$ is the updated classifier model where, for every $s' \in S$:

$$f^{c:=\varphi}(s') = \begin{cases} c \text{ if } (C, s') \models \varphi, \\ f(s') \text{ otherwise.} \end{cases}$$

Intuitively, the operation $c{:=}\varphi$ consists in globally classifying all instances satisfying φ with value c.

Dynamic operators $[c := \varphi]$ are useful for modeling a classifier's revision. Specifically, new knowledge can be injected into the classifier thereby leading to a change in its classification. For example, the classifier could learn that if an

object is a furniture, has one or more legs and has a flat top, then it is a table. This is captured by the following assignment:

$$\texttt{table} := \mathit{objIsFurniture} \wedge \mathit{objHasLegs} \wedge \mathit{objHasFlatTop}.$$

An application of dynamic change is to model the training process of a classifier, together with counterfactual conditionals with "?" in Section 4. Suppose at the beginning we have a CM $C = (S, f)$ which is totally ignorant, i.e., $\forall s \in S, f(s) =?$. We then prepare to train the classifier. The training set consists of pairs $(s_1, x_1), (s_2, x_2) \dots (s_n, x_n)$ where $\forall i \in \{1, \dots, n\}$, $s_i \in S, x_i \in (\mathit{Val} \backslash \{?\})$ and $\forall j \in \{1, \dots, n\}$, $i \neq j$ implies $s_i \neq s_j$. We train the classifier by revising it with $[x_1 := \widehat{s}_1] \dots [x_n := \widehat{s}_n]$ one by one. Obviously the order does not matter here. In other words, we re-classify some states. With a bit abuse of notation, let $C^{train} = (S, f^{train})$ denote the model resulting from the series of revisions. We finish training by inducing the final model $C^{\dagger} = (S, f^{\dagger})$ from C^{train}, where $\forall s \in S, f^{\dagger}(s) = c$, if $(C^{train}, s) \models \mathsf{apprDec}(\mathit{Atm}_0, c)$, otherwise $f^{\dagger}(s) = f^{train}(s)$. This is an example of modeling a special case of the so-called *k-nearest neighbour (KNN) classification* in machine learning [10], where the distance is measured by cardinality. If a new case/instance has to be classified, we see how the most similar cases to the new case were classified. If all of them (k of them in the case of KNN) were classified using the same category, we put the new case into that category.

The logics $\mathsf{BCL}{-}\mathsf{DC}$ and $\mathsf{WBCL}{-}\mathsf{DC}$ (BCL and WBCL with Decision Change) extend the logic BCL and WBCL by the dynamic operators $[c{:=}\varphi]$. They are defined as follows.

Definition 16 (Logics $\mathsf{BCL}{-}\mathsf{DC}$ and $\mathsf{WBCL}{-}\mathsf{DC}$)**.** *We define* $\mathsf{BCL}{-}\mathsf{DC}$ *(resp.* $\mathsf{WBCL}{-}\mathsf{DC}$*) to be the extension of* BCL *(resp.* WBCL*) of Definition 7 (resp. Definition 8) generated by the following reduction axioms for the dynamic operators* $[c{:=}\varphi]$*:*

$$\begin{aligned}
[c{:=}\varphi]\mathsf{t}(c) &\leftrightarrow \big(\varphi \vee \mathsf{t}(c)\big)\\
[c{:=}\varphi]\mathsf{t}(c') &\leftrightarrow \big(\neg\varphi \wedge \mathsf{t}(c')\big) \ \textit{if } c \neq c'\\
[c{:=}\varphi]p &\leftrightarrow p\\
[c{:=}\varphi]\neg\psi &\leftrightarrow \neg[c{:=}\varphi]\psi\\
[c{:=}\varphi](\psi_1 \wedge \psi_2) &\leftrightarrow \big([c{:=}\varphi]\psi_1 \wedge [c{:=}\varphi]\psi_2\big)\\
[c{:=}\varphi][X]\psi &\leftrightarrow [X][c{:=}\varphi]\psi
\end{aligned}$$

and the following rule of inference:

$$\frac{\varphi_1 \leftrightarrow \varphi_2}{\psi \leftrightarrow \psi[\varphi_1/\varphi_2]} \qquad (\mathbf{RE})$$

It is routine exercise to verify that the equivalences in Definition 16 are valid for the class **CM** and that the rule of replacement of equivalents (**RE**) preserves validity. The completeness of $\mathsf{BCL}{-}\mathsf{DC}$ (resp. $\mathsf{WBCL}{-}\mathsf{DC}$) for this

class of models under the finite-variable assumptions (resp. infinite-variable assumption) follows from Corollary 1 (resp. Corollary 2), in view of the fact that the reduction axioms and the rule of replacement of proved equivalents can be used to find, for any $\mathcal{L}^{dyn}$-formula, a provably equivalent $\mathcal{L}$-formula.

Theorem 10. *Let Atm_0 be finite. Then, the logic* $\mathsf{BCL}-\mathsf{DC}$ *is sound and complete relative to the class* **CM**.

Theorem 11. *Let Atm_0 be countably infinite. Then, the logic* $\mathsf{WBCL}-\mathsf{DC}$ *is sound and complete relative to the class* **CM**.

The following complexity results are consequences of Theorems 7 and 8 and the fact that via the reduction axioms in Definition 8 we can find a polynomial reduction of satisfiability checking for formulas in $\mathcal{L}^{dyn}$ to satisfiability checking for formulas in $\mathcal{L}$.

Theorem 12. *Let Atm_0 be finite and fixed. Then, checking satisfiability of formulas in $\mathcal{L}^{dyn}(Atm)$ relative to* **CM** *can be done in polynomial time.*

Theorem 13. *Let Atm_0 be countably infinite. Then, checking satisfiability of formulas in $\mathcal{L}^{dyn}(Atm)$ relative to* **CM** *is NEXPTIME-complete.*

6.2 Epistemic Extension

In the second extension we suppose that a classifier is an agent which has to classify what it perceives. The agent could have uncertainty about the actual instance to be classified since it cannot see all its input features.

In order to represent the agent's epistemic state and uncertainty, we introduce an epistemic modality of the form K which is used to represent what the agent knows in the light of what it sees. Similar notions of visibility-based knowledge can be found in [8, 45, 21, 44].

The language for our epistemic extension is noted $\mathcal{L}^{epi}(Atm)$ and defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi \mid \mathsf{K}\varphi,$$

where p ranges over Atm_0, c ranges over Val, and $X \subseteq^{\text{fin}} Atm_0$.

In order to interpret the new modality K, we have to enrich classifier models with an epistemic component.

Definition 17 (Epistemic classifier model)**.** *An epistemic classifier model (ECM) is a tuple $E = (S, f, Obs)$ where $C = (S, f)$ is a classifier model and $Obs \subseteq Atm_0$ is the set of atomic propositions that are visible to the agent. The class of ECMs is noted* **ECM**.

Given an ECM $E = (S, f, Obs)$, we can define an epistemic indistinguishability relation which represents the agent's uncertainty about the actual input instance.

Definition 18 (Epistemic indistinguishability relation)**.** *Let $E = (S, f, Obs)$ be an ECM. Then, $\sim$ is the binary relation on S such that, for all $s, s' \in S$:*

$$s \sim s' \text{ if and only if } (s \cap Obs) = (s' \cap Obs).$$

Clearly, the relation $\sim$ so defined is an equivalence relation. According to the previous definition, the agent cannot distinguish between two states s and s', noted $s \sim s'$, if and only if the truth values of the visible variables are the same at s and s'.

The interpretation for formulas in $\mathcal{L}^{epi}(Atm)$ extends the interpretation for formulas in $\mathcal{L}(Atm)$ given in Definition 2 by the following condition for the epistemic operator:

$$(E, s) \models \mathsf{K}\varphi \iff \forall s' \in S : \text{ if } s \sim s' \text{ then } (E, s') \models \varphi.$$

As the following theorem indicates, the complexity result of Section 3.2 for the finite-variable case generalizes to the epistemic extension.

Theorem 14. *Let Atm_0 be finite. Then, checking satisfiability of formulas in $\mathcal{L}^{epi}(Atm)$ relative to* **ECM** *can be done in polynomial time.*

In order to illustrate the intuition behind the epistemic modality K we go back to the example of the application for a loan to a bank.

Example 6. *Suppose the application is submitted through an online system which has to automatically decide whether it is acceptable or not. In his/her application, an applicant has to specify a value for each feature. Moreover, suppose the system receives an incomplete application: the applicant has only indicated that she is female, owns an apartment and lives in the city center, but she has forgotten to specify whether she has an employment or not. In this case, the value of the employement variable is not "visible" to the system. In formal terms, we extend the CM given in Example 2 by the visibility set $Obs = \{male, center, owner\}$ to obtain a ECM $E = (S, f, Obs)$. It is easy to check that the following holds:*

$$\big(E, \{center, employed, owner\}\big) \models \neg\mathsf{K}\ \mathsf{t}(0) \wedge \neg\mathsf{K}\ \mathsf{t}(1).$$

This means that, on the basis of its partial knowledge of the applicant's identity, the system does not know what to decide.

However, the system knows that if turns out that the applicant is employed then its application should be accepted:

$$\big(E, \{center, employed, owner\}\big) \models \mathsf{K}\big(employed \rightarrow \mathsf{t}(1)\big).$$

Finally, the classifier knows that if turns out that the applicant is employed, then the fact that she is employed and that she owns a property will abductively explain the decision to accept her application:

$$\big(E, \{center, employed, owner\}\big) \models \mathsf{K}\big(employed \rightarrow \mathsf{AXp}(employed \wedge owner, 1)\big).$$

7 Conclusion

We have introduced a modal language and a formal semantics that enable us to capture the *ceteris paribus* nature of binary classifiers. We have formalized in the language a variety of notions which are relevant for understanding a classifier's behavior including counterfactual conditional, abductive and contrastive explanation, bias. We have provided two extensions that support reasoning about classifier change and a classifier's uncertainty about the actual instance to be classified. We have also offered axiomatics and complexity results for our logical setting.

We believe that the complexity results presented in the paper are exploitable in practice. We have shown that satisfiability checking in the basic setting and in its dynamic and epistemic extension is polynomial when finitely many variables are assumed. In the infinite-variable setting, it becomes NEXPTIME-complete and NP-complete when restricting to the language in which the only primitive modal operator is the universal modality $[\emptyset]$. In future work, we plan (i) to find a number of satisfiability preserving translations from our modal languages to the modal logic S5 and then from S5 to propositional logic using existing techniques [7], and (ii) to exploit SAT solvers for automated verification and generation of explanations and biases in binary classifiers.

Another direction of future research is the generalization of the epistemic extension given in Section 6.2 to the multi-agent case. The idea is to conceive classifiers as agents and to be able to represent both the agents' uncertainty about the instance to be classified and their knowledge and uncertainty about other agents' knowledge and uncertainty (i.e., higher-order knowledge and uncertainty). Similarly, we plan to investigate more in depth classifier dynamics we briefly discussed in Section 6.1. The idea is to see them as learning dynamics. Based on this idea, we plan to study the problem of finding a sequence of update operations guaranteeing that the classifier will be able to make approximate decisions for a given set of instances.

Finally, all classifiers we handle in this paper are essentially "white box", in the sense that we have perfect knowledge of them, so that we can compute their explanations. However, "black box" classifiers are the most interesting ones to XAI. In [32] we conceived a "black box" classifier as an agent's uncertainty among many possible "white box" classifiers. We represented it by extending our language with a modal operator ranging over all possible functions which are compatible with the agent's partial knowledge. All notions of explanation we defined in this paper can be generalized to the "black box" setting. However, there are some important differences between the two settings. For instance, in a "black box" classifier AXp does not always exist, as we showed in [32], which contradicts Proposition 6.

Acknwoledgments

This work is supported by the ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANITI).

A Tecnical annex

This technical annex contains a selection of proofs of the results given in the paper.

A.1 Proof of Proposition 2

Proof. Suppose C is X-definite but $(C, s) \models \neg\mathsf{Defin}(X)$, which means that $\exists c \in Val$ s.t. $(C, s) \models \neg =(X, \mathsf{t}(c))$. W.l.o.g., we assume that $(C, s) \models \neg[\emptyset](\neg\mathsf{t}(c) \to [X]\neg\mathsf{t}(c))$. That is to say, $\exists s' \in S$, s.t. $f(s') \neq c$ but $(C, s') \models \langle X\rangle\mathsf{t}(c)$. The latter indicates that $\exists s'' \in S$, s.t. $s'' \cap X = s' \cap X$ but $f(s'') = c$, which violates X-definiteness.

Let $(C, s) \models \mathsf{Defin}(X)$, and assume $f(s) = c$ Then since $(C, s) \models [\emptyset](\mathsf{t}(c) \to [X]\mathsf{t}(c))$, we have $\forall s' \in S$ if $s' \cap X = s \cap X$ then $f(s') = c = f(s)$, which is what X-definiteness says. □

A.2 Proof of Theorem 1

Proof. For the left to right direction, given a CM $C = (S, f)$ and $s_0 \in S$ s.t. $(C, s_0) \models \varphi$, we construct a DM $M^\flat = (W^\flat, (\equiv^\flat_X)_{X \subseteq^{\text{fin}} Atm_0}, V^\flat)$ as follows

- $W^\flat = S$
- $s \equiv^\flat_X s'$ if $s \cap X = s' \cap X$
- $V^\flat(s) = s \cup \{\mathsf{t}(f(s))\}$.

It is easy to check that $M^\flat$ is indeed a DM and $(M^\flat, s_0) \models \varphi$.

For the other direction, given a DM $\big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ and $w_0 \in W$ s.t. $(M, w_0) \models \varphi$, we construct a CM $C^\sharp = (S^\sharp, f^\sharp)$ as follows

- $S^\sharp = \{V_{Atm_0}(w) : w \in W\}$
- $\forall V_{Atm_0}(w) \in S^\sharp, f^\sharp(V_{Atm_0}(w)) = c$, if $V_{Dec}(w) = \{\mathsf{t}(c)\}$.

It is routine to check that $C^\sharp$ is a CM, and $(C^\sharp, V_{Atm_0}(w_0)) \models \varphi$.

□

A.3 Proof of Theorem 2

Proof. The proof is conducted by constructing the canonical model.

Definition 19 (Theory)**.** *A set of formulas* Γ *is said to be a* BCL-theory *if it contains all theorems of* BCL *and is closed under* ***MP*** *and* $\boldsymbol{Nec}_{[\emptyset]}$*. It is said to be a* consistent BCL*-theory if it is a theory and* $\bot \notin \Gamma$*. It is said to be a* maximal consistent BCL*-theory (MCT for short), if it is a consistent theory and for all consistent theory* Γ'*, if* $\Gamma \subseteq \Gamma'$ *then* $\Gamma = \Gamma'$*.*

Lemma 1 (Lindenbaum-type)**.** *Let* Δ *be a consistent* BCL*-theory and* $\varphi \notin \Delta$ *Then, there is a maximal consistent* BCL*-theory* Γ *s.t.* $\Delta \subseteq \Gamma$ *and* $\varphi \notin \Gamma$*.*

The proof is standard and omitted (see, e.g. [5, p. 197]).

Definition 20 (Canonical model)**.** *The canonical decision model* $\mathfrak{M} = (W^c, (\equiv^c_X)_{X \subseteq^{\text{fin}} Atm_0}, V^c)$ *is defined as follows*

- $W^c = \{\Gamma : \Gamma$ *is a maximal consistent* BCL *theory.*$\}$
- $\Gamma \equiv^c_X \Delta \iff \{[X]\varphi : [X]\varphi \in \Gamma\} = \{[X]\varphi : [X]\varphi \in \Delta\}$
- $V^c(\Gamma) = \{p : p \in \Gamma\}$

We omit the superscript ^{c} whenever there is no misunderstanding.

Lemma 2. *Let* Γ *be an MCT. Then* $[X]\varphi \to \varphi \in \Gamma$*.*

Proof. Suppose $[X]\varphi \to \varphi \notin \Gamma$, then by the maximality of Γ and $\mathbf{Red}_{[\emptyset]}$, we have $\bigwedge_{Y \subseteq X} \left(\mathsf{cn}_{Y,X} \to [\emptyset](\mathsf{cn}_{Y,X} \to \varphi)\right) \wedge \neg\varphi \in \Gamma$. Since Γ is maximally consistent, there is exactly one $Z \subseteq X$ s.t. $\mathsf{cn}_{Z,X} \in \Gamma$. By **MP** we have $[\emptyset](\mathsf{cn}_{Z,X} \to \varphi) \in \Gamma$, and by $\mathbf{K}_{[\emptyset]}$ and **MP** we have $\varphi \in \Gamma$. But than Γ is inconsistent, since $\varphi \wedge \neg\varphi \in \Gamma$. Hence the supposition fails, which means $[X]\varphi \to \varphi \in \Gamma$. □

Lemma 3. *The canonical model* $\mathfrak{M}$ *is indeed a decision model.*

Proof. Check the conditions one by one. For **C1**, we need show $\Gamma \equiv^c_X \Delta$, if $\forall p, p \in V(\Gamma) \cap X$ implies $p \in V(\Delta)$. Suppose not, then w.l.o.g. we have some $q \in V(\Gamma) \cap X, q \notin V(\Delta)$, by maximality of Δ namely $\neg q \in \Delta$. However, we have $[q]q \in \Gamma$, for $q \to [q]q$ is a theorem, and by definition of $\equiv^c_X, [q]q \in \Delta$, hence $q \in \Delta$, since $[q]q \to q \in \Delta$. But now we have a contradiction. **C2-4** hold obviously due to axioms **AtLeast**, **AtMost**, **Def** and **Funct** respectively. □

Lemma 4 (Existence)**.** *Let* $\mathfrak{M} = (W^c, (\equiv^c_X)_{X \subseteq^{\text{fin}} Atm_0}, V^c)$ *be the canonical model,* Γ *be an MCT. Then, if* $\langle\emptyset\rangle\varphi \in \Gamma$ *then* $\exists \Gamma' \in W^c$ *s.t.* $\Gamma \equiv^c_\emptyset \Gamma'$ *and* $\varphi \in \Gamma'$*.*

The proof is following the same line in e.g. [5, p. 198-199] and omitted.

Lemma 5 (Truth)**.** *Let* $\mathfrak{M}$ *be the canonical model,* Γ *be an MCT,* $\varphi \in \mathcal{L}(Atm_0)$*. Then* $\mathfrak{M}, \Gamma \models \varphi \iff \varphi \in \Gamma$*.*

Proof. By induction on φ. We only show the interesting case when φ takes the form $[X]\psi$.

For $\Longleftarrow$ direction, if $[X]\psi \in \Gamma$, since for any $\Delta \equiv_X \Gamma$, $[X]\psi \in \Delta$, then thanks to $[X]\psi \to \psi \in \Delta$ we have $\psi \in \Delta$. By induction hypothesis this means $\Delta \models \psi$, therefore $\Gamma \models [X]\psi$.

For $\Longrightarrow$ direction, suppose not, namely $[X]\psi \notin \Gamma$. Then consider a theory $\Gamma' = \{\neg\psi\} \cup \{[X]\chi : [X]\chi \in \Gamma\}$. It is consistent since $\psi \notin \Gamma$. Then take any $\Delta \in W$ s.t. $\Gamma' \subseteq \Delta$. We have $\Delta \equiv_X \Gamma$, but $\Delta \not\models \psi$ by induction hypothesis. However, this contradicts $\Gamma \models [X]\psi$. □

Now the completeness of **DM** w.r.t. BCL is a corollary of Lemma 3 and 5. □

A.4 Proof of Theorem 3

Proof. Let $\big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ be a QDM and $w_0 \in W$ s.t. $(M, w_0) \models \varphi$. Let $sf(\varphi)$ be the set of all subformulas of φ and let $sf^+(\varphi) = sf(\varphi) \cup Dec$. Moreover, $\forall v, u \in W$, we define $v \simeq u \iff \forall \psi \in sf^+(\varphi)$, $(M, v) \models \psi$ iff $(M, u) \models \psi$. Finally, we define $[v] = \{u \in W : v \simeq u\}$.

Now we construct a filtration through $sf^+(\varphi)$, $M' = (W', (\equiv'_X)_{X \subseteq^{\text{fin}} Atm_0}, V')$ as follows

- $W' = \{[v] : v \in W\}$
- $\forall X \subseteq^{\text{fin}} Atm_0$, $[v] \equiv'_X [u]$, iff $V'_X([v]) = V'_X([u])$
- $V'([v]) = V_{sf^+(\varphi) \cap Atm_0}(v)$

M' is indeed a filtration. We need show that it satisfies the two conditions.

1) $v \equiv_X u \iff V_X(v) = V_X(u) \Longrightarrow V'_X([v]) = V'_X([u]) \iff [v] \equiv'_X [u]$. Suppose $v \equiv_X u$. By construction of V', $\forall p \in X \cap sf^+(\varphi), p \in V'_X([v]) p \in V(v) \iff p \in V(u) \iff p \in V'_X([u])$, and $\forall p \in X \setminus sf^+(\varphi), p \notin V'_X([v])$ and $p \notin V'_X([u])$. As a result, $V'_X([v]) = V'_X([u])$ which means $[v] \equiv'_X [u]$.

2) If $[v] \equiv'_X [u]$, then $\forall [X]\psi \in sf^+(\varphi)$: if $(M, v) \models [X]\psi$ then $(M, u) \models \psi$. The crucial point is that $\forall v, v' \in [v], \forall u, u' \in [u]$, $\forall [X]\psi \in sf^+(\varphi)$, if $[v] \equiv'_X [u]$, then $v \equiv_X v' \equiv_X u \equiv_X u'$ by the definitions of V' and $\simeq$. Hence by satisfaction relation of M we have if $(M, v) \models [X]\psi$ then $(M, u) \models \psi$.

Moreover, M' is a finite-QDM. For **C1** it is given as the definition of V'. **C2** and **C3** hold because of $sf^+(\varphi) = sf(\varphi) \cup Dec$.

Finally, we need prove $(M, w_0) \models \varphi$ iff $(M', [w_0]) \models \varphi$. We only show when φ takes the form $[X]\psi$. Given $(M, w_0) \models [X]\psi$, i.e. $\forall v \in W$, if $w_0 \equiv_X v$ then $(M, v) \models \psi$. By definitions of $\equiv'_X$ and **C1** we have $V'_X([w_0]) = V'_X([v])$, by induction hypothesis $(M', [v]) \models \psi$, which means $(M', [w_0]) \models [X]\psi$. If $(M', [w_0]) \models [X]\psi$, i.e. $\forall [v] \in W'$, if $[v] \equiv'_X [w_0]$ then $(M', [v]) \models \psi$. Then by definitions of V' and $\simeq$ we have $w_0 \equiv_X v$, by induction hypothesis $(M, v) \models \psi$. □

A.5 Proof of Theorem 4

Proof. The right to left direction is obvious since any finite-DM is a finite-QDM. For the other direction, suppose there is a finite-QDM $\big(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V\big)$ and $w \in W$ s.t. $(M, w) \models \varphi$. Since Atm_0 is infinite, we can construct an injection $\iota : W \longrightarrow Atm_0 \setminus Atm(\varphi)$. Then, we construct a finite-DM $M' = (W', (\equiv'_X)_{X \subseteq^{\text{fin}} Atm_0}, V')$ as follows

- $W' = W$
- $w \equiv'_X v$ iff $V'_X(w) = V'_X(v)$
- $V'(w) = (V(w) \cup \{\iota(w)\}) \setminus \{p : \exists v \in W, v \neq w \ \&\ \iota(v) = p\}$.

It is easy to check that M' is indeed a finite-DM. By induction we show that $(M', w) \models \varphi$. When φ is some p, we have $V(w) = V'(w)$ since the injection ι has nothing to do with φ. The case of $\mathtt{t}(c)$ is the same. The Boolean cases are straightforward. Finally when φ takes form $[X]\psi$. Again since ι does not change valuation in φ, we have $\forall v \in W, V_X(v) = V'_X(v)$. Hence we have $(M, w) \models [X]\psi \iff \forall v \in W$, if $V_X(w) = V_X(v)$ then $(M, v) \models \psi \iff \forall v \in W$, if $V'_X(w) = V'_X(v)$ then $(M', v) \models \psi \iff (M', w) \models [X]\psi$. □

A.6 Proof of Theorem 7

Proof. Suppose Atm_0 is finite and fixed. In order to determine whether a formula φ is satisfiable for the class **CM**, we are going to verify whether φ is satisfied in each CM, by doing this sequentially one CM after the other. The corresponding algorithm runs in polynomial time in the size of φ since: (i) there is a finite, constant number of CMs and (ii) model checking for the language $\mathcal{L}(Atm)$ relative to a pointed CM is polynomial. This means that, when Atm_0 is finite and fixed, satisfiability checking has the same complexity as model checking. Regarding (i), the finite, constant number of CMs in the class **CM** is $\sum_{S \subseteq 2^{Atm_0}} |Val|^{|S|}$. Indeed, for every $S \subseteq 2^{Atm_0}$, we consider the number of functions from S to Val. Regarding (ii), it is easy to build a model checking algorithm running in polynomial time. It is sufficient to adapt the well-known "labelling" model checking algorithm for the basic multimodal logics and CTL [9]. The general idea of the algorithm is to label the states of a finite model step-by-step with sub-formulas of the formula φ to be checked, starting from the smallest ones, the atomic propositions appearing in φ. At each step, a formula should be added as a label to just those states of the model at which it is true. □

A.7 Proof of Theorem 8

Proof. As for NEXPTIME-hardness, in [17] the following *ceteris paribus* modal language, noted $\mathcal{L}_{\mathsf{CP}}(Prop)$, is considered with $Prop$ a countable set of atomic propositions:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi,$$

where p ranges over *Prop* and X is a finite set of atomic propositions from *Prop*. Formulas for this language are interpreted relative to a *simple model* $S \subseteq 2^{Atm_0}$ and a state $s \in S$ in the expected way as follows (we omit boolean cases since they are interpreted in the usual way): $(S, s) \models p$ iff $p \in s$; $(S, s) \models [X]\varphi$ iff $\forall s' \in S$: if $s \cap X = s' \cap X$ then $(S, s') \models \varphi$. It is proved that, when *Prop* is countably infinite, satisfiability checking for formulas in $\mathcal{L}_{\mathsf{CP}}(Prop)$ relative to the class **SM** of simple models is NEXPTIME-hard [17, Lemma 2 and Corollary 2]. It follows that satisfiability checking for formulas in our language $\mathcal{L}(Atm)$ with Atm_0 countably infinite is NEXPTIME-hard too.

As for membership, let tr be the following translation from the language $\mathcal{L}(Atm)$ to the language $\mathcal{L}_{\mathsf{CP}}\big(Atm_0 \cup \{p_{\mathsf{t}(c)} : c \in Val\}\big)$:

$$
\begin{aligned}
&tr(p) = p,\\
&tr(\mathsf{t}(c)) = p_{\mathsf{t}(c)},\\
&tr(\neg\varphi) = \neg tr(\varphi),\\
&tr(\varphi \wedge \psi) = tr(\varphi) \wedge tr(\psi),\\
&tr([X]\varphi) = [X]tr(\varphi).
\end{aligned}
$$

By induction on the structure of φ, it is routine to verify that $\varphi \in \mathcal{L}(Atm)$ is satisfiable for the class **QDM** of Definition 9 if and only if $[\emptyset]\big(\varphi_1 \wedge \varphi_2\big) \wedge tr(\varphi)$ is satisfiable for the class **SM** of simple models, with

$$
\begin{aligned}
\varphi_1 &=_{def} \bigvee_{c \in Val} p_{\mathsf{t}(c)},\\
\varphi_2 &=_{def} \bigwedge_{c,c' \in Val : c \neq c'} \big(p_{\mathsf{t}(c)} \rightarrow \neg p_{\mathsf{t}(c')}\big).
\end{aligned}
$$

Hence, by Theorem 5 we have that, when Atm_0 is countably infinite, $\varphi \in \mathcal{L}(Atm)$ is satisfiable for the class **CM** of classifier models if and only if $[\emptyset]\big(\varphi_1 \wedge \varphi_2\big) \wedge tr(\varphi)$ is satisfiable for the class **SM** of simple models. Since the translation tr is linear and satisfiability checking for formulas in $\mathcal{L}_{\mathsf{CP}}\big(Atm_0 \cup \{p_{\mathsf{t}(c)} : c \in Val\}\big)$ relative to the class **SM** is in NEXPTIME in the infinite-variable case [17, Lemma 2 and Corollary 1], checking satisfiability of formulas in $\mathcal{L}(Atm)$ relative to the class **CM** is in NEXPTIME too, with Atm_0 countably infinite. □

A.8 Proof of Theorem 9

Proof. NP-hardness follows from the NP-harndess of propositional logic.

In order to prove NP-membership, we can use the translation given in the proof of Theorem 8 to give a polynomial reduction of satisfiability checking of formulas in $\mathcal{L}^{\{[\emptyset]\}}(Atm)$ relative to **CM** to satisfiability checking in the modal logic S5. The latter problem is known to be in NP in the infinite-variable case [26]. □

A.9 Proof of Proposition 3

Proof. For the right direction, we have $closest_C(s,\varphi,X) \subseteq ||\psi||_C$ from the antecedent. Suppose towards a contradiction that the consequent does not hold. Then, $\exists k \in \{0,\dots,|X|\}, Y_1, Y_2 \subseteq X$ with $|Y_1| = |Y_2| = k$, s.t. $(C,s) \models \langle Y_1\rangle\varphi \wedge \bigwedge_{Y\subseteq X: k<|Y|}[Y]\neg\varphi \wedge \langle Y_2\rangle(\varphi \wedge \neg\psi)$. The last conjunct means that $\exists s' \in S, s' \cap X = s \cap X = Y_2$ and $(C,s') \models \varphi \wedge \neg\psi$. But the conjuncts together guarantee that $s' \in closest_C(s,\varphi,X)$, because $sim_C(s,s',X) = k$, and it is an argmax by definition of $closest_C(s,\varphi,X)$. It is the desired contradiction, since $s' \notin ||\psi||_C$.

For the other direction, we need show $closest_C(s,\varphi,X) \subseteq ||\psi||_C$, given the antecedent. Suppose the opposite towards a contradiction. Then by definition, $\exists s^* \in closest_C(s,\varphi,X), s' \notin ||\psi||_C$. Let $s \cap X = s^* \cap X = Y^*$, and $sim_C(s,s',X) = k^*$. Then we have $(C,s) \models \mathsf{maxSim}(\varphi,X,k^*) \wedge \langle Y^*\rangle(\varphi \wedge \neg\psi)$, which contradicts the antecedent. To see that, notice the second conjunct is because of $(C,s^*) \models \varphi \wedge \neg\psi$, and the first conjunct because of $sim_C(s,s^*,X) = k^*$ and $s^* \in closest_C(s,\varphi,X)$. □

A.10 Proof of Proposition 4

Proof. The first validity is obvious, since if $closest_C(s,\varphi,X) \subseteq ||\mathsf{t}(c)||_C$ then $closest_C(s,\varphi,X) \nsubseteq ||\mathsf{t}(c')||_C$ given $c' \neq c$. For the second validity, notice that $\{s\} = closest_C(s,\varphi,Atm_0)$, if $(C,s) \models \varphi$. Hence if $(C,s) \models \mathsf{t}(c)$, then we have $closest_C(s, \bigvee_{c'\in Val: c'\neq ?}, Atm_0) = \{s\} \subseteq ||\mathsf{t}(c)||_C$. □

A.11 Proof of Proposition 5

Proof. Let (C,s) be a pointed CM and $(C,s) \models \mathsf{AXp}(\lambda,c)$, which directly gives us $(C,s) \models \lambda$. Now since λ is an implicant of c, $(C,s) \models [Atm(\lambda)]\mathsf{t}(c)$, for otherwise $\exists s'$, s.t. $(C,s') \models \lambda \wedge \neg\mathsf{t}(c)$; and since λ is prime, we have $(C,s) \models \bigwedge_{p\in Atm(\lambda)}\langle Atm(\lambda) \setminus \{p\}\rangle\neg\mathsf{t}(c))$, otherwise $\exists\lambda'$, s.t. $\lambda' \subset \lambda$ and λ' is also an implicant of c. The other direction is proven in the same way and omitted. □

A.12 Proof of Proposition 6

Proof. Suppose towards a contradiction that C is finitely-definite, but $\exists c \in Val$, s.t. $\forall\lambda \in Term$, if $(C,s) \models \lambda$ then $(C,s) \models \neg\mathsf{PImp}(\lambda,c)$. That is to say, $\exists s_1 \in S$ s.t. $(M,s_1) \models \lambda$ but either $f(s_1) \neq c$ or $\exists s_2 \in S$ s.t. $\exists p \in Atm(\lambda)$, $s_1 \cap (Atm(\lambda\setminus\{p\}) = s_2 \cap (Atm(\lambda\setminus\{p\})$ but $f(s_2) \neq c$. Hence C is neither $Atm(\lambda)$-definite nor $(Atm(\lambda \setminus \{p\})$-definite. Either case C is not finitely-definite, since λ is arbitrarily selected from $Term$. □

A.13 Proof of Proposition 7

Proof. For the first validity, let $C = (S,f) \in \mathbf{CM}$ and $s \in S$ and suppose $(C,s) \models \mathsf{CXp}(\lambda,c)$. By definition of $\mathsf{CXp}(\lambda,c)$ we have $(C,s) \models \mathsf{t}(c)$. We need

to show $(C, s) \models \overline{\lambda} \Rightarrow \neg \mathsf{t}(c)$. By the antecedent, $\exists s' \in S$, s.t. $s \triangle s' = Atm(\lambda)$ and $f(s') \neq c$. It is not hard to show that $closest_C(s, \overline{\lambda}, Atm) = \{s'\}$. Therefore $(C, s) \models \overline{\lambda} \Rightarrow \neg \mathsf{t}(c)$, since $closest_C(s, \overline{\lambda}, Atm_0) \subseteq ||\neg \mathsf{t}(c)||_C$. For the second validity, the right direction is a special case of the first validity. To show the left direction, from the antecedent we have $\exists s' \in S$, s.t. $s' \triangle s = Atm(l)$ and $\{s'\} = closest_C(s, l, Atm_0)$. Hence $(C, s) \models l \wedge \langle Atm_0 \setminus Atm(l) \rangle \neg \mathsf{t}(c) \wedge [Atm_0] \mathsf{t}(c)$, which is by definition $(C, s) \models \mathsf{CXp}(l, c)$. □

A.14 Proof of Proposition 8

Proof. We show that for any $C = (S, f) \in \mathbf{CM}$, both directions are satisfied in (C, s) for some $s \in S$. The right to left direction is obvious, since from the antecedent we know there is a property λ' s.t. $\exists s' \in S, s \triangle s' = Atm(\lambda') \subseteq \mathsf{PF}$ and $(C, s') \models \neg \mathsf{t}(c)$, which means $(C, s) \models \mathsf{Bias}(c)$. The other direction is proven by contraposition. Suppose for any λ s.t. $Atm(\lambda) \subseteq \mathsf{PF}$, $(C, s) \models \neg \mathsf{CXp}(\lambda, c)$, then it means $\forall s' \in S$, if $s \triangle s' = Atm(\lambda)$, then $f(s') = c$, which means $(C, s) \models \neg \mathsf{Bias}(c)$. □

A.15 Proof of Theorem 14

Proof. Suppose $|Atm_0|$ is finite. As in the proof of Theorem 7, we can show that the size of the model class **ECM** is bounded by some fixed integer. Thus, in order to determine whether a formula φ $\mathcal{L}^{epi}(Atm)$ is satisfiable for this class, it is sufficient to repeat model checking a number of times which is bounded by some integer. Model checking for the language $\mathcal{L}^{epi}(Atm)$ with respect to a pointed ECM is polynomial. □

References

[1] Leila Amgoud and Jonathan Ben-Naim. Axiomatic foundations of explainability. In *31st International Joint Conference on Artificial Intelligence (IJCAI 2022)*, 2022.

[2] Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the computational intelligibility of boolean classifiers. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 18, pages 74–86, 2021.

[3] Alexandru Baltag and Johan van Benthem. A simple logic of functional dependence. *Journal of Philosophical Logic*, 50(5):939–1005, 2021.

[4] Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8(1), pages 8–13, 2017.

[5] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, Cambridge, Massachusetts, 2001.

[6] Alexander Borgida. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems (TODS)*, 10(4):565–603, 1985.

[7] Thomas Caridroit, Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima, and Valentin Montmirail. A SAT-based approach for solving the modal logic s5-satisfiability problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3864–3870. AAAI Press, 2017.

[8] Tristan Charrier, Andreas Herzig, Emiliano Lorini, Faustine Maffre, and François Schwarzentruber. Building epistemic logic from observations and public announcements. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 268–277. AAAI Press, 2016.

[9] Edmund M. Clarke and Bernd-Holger Schlingloff. Model checking. In Alan J. A. Robinson and Andrei Voronkov, editors, *Handbook of automated reasoning*, pages 1635–1790. Elsevier, 2001.

[10] Padraig Cunningham and Sarah Jane Delany. K-nearest neighbour classifiers - a tutorial. *ACM Computing Surveys*, 54(6):1–25, 2022.

[11] Mukesh Dalal. Investigations into a theory of knowledge base revision: preliminary report. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, volume 2, pages 475–479. Citeseer, 1988.

[12] Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In *24th European Conference on Artificial Intelligence (ECAI 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 712–720. IOS Press, 2020.

[13] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603, 2018.

[14] Ronald Fagin, Yoram Moses, Joseph Y Halpern, and Moshe Y Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[15] Patrick Girard and Marcus Anthony Triplett. Ceteris paribus logic in counterfactual reasoning. In *Proceedings of the Fifteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015)*, pages 176–193, 2016.

[16] Nelson Goodman. *Fact, fiction, and forecast*. Harvard University Press, 1955.

[17] Davide Grossi, Emiliano Lorini, and François Schwarzentruber. The ceteris paribus structure of logics of game forms. *Journal of Artificial Intelligence Research*, 53:91–126, 2015.

[18] Joseph Y. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(2):361–372, 1995.

[19] Joseph Y Halpern. *Actual causality*. MiT Press, 2016.

[20] Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of science*, 15(2):135–175, 1948.

[21] Andreas Herzig, Emiliano Lorini, and Faustine Maffre. A poor man's epistemic logic based on propositional assignment and higher-order observation. In *Proceedings of the 5th International Workshop on Logic, Rationality, and Interaction*, Lecture Notes in Computer Science, pages 156–168. Springer, 2015.

[22] Alexey Ignatiev, Martin C. Cooper, Mohamed Siala, Emmanuel Hebrard, and Joao Marques-Silva. Towards formal fairness in machine learning. In *International Conference on Principles and Practice of Constraint Programming*, pages 846–867. Springer, 2020.

[23] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and Joao Marques-Silva. From contrastive to abductive explanations and back again. In *International Conference of the Italian Association for Artificial Intelligence*, pages 335–355. Springer, 2020.

[24] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of the Thirty-third AAAI Conference on Artificial Intelligence (AAAI-19)*, volume 33, pages 1511–1519, 2019.

[25] Boris Kment. Counterfactuals and explanation. *Mind*, 115(458):261–310, 2006.

[26] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM journal on computing*, 6(3):467–480, 1977.

[27] David K. Lewis. *Counterfactuals*. Harvard University Press, 1973.

[28] David K. Lewis. Counterfactual dependence and time's arrow. *Noûs*, pages 455–476, 1979.

[29] David K. Lewis. Causal explanation. In *Philosophical Papers*, volume 2, pages 214–240. Oxford University Press, 1986.

[30] David K. Lewis. Causation. *Journal of Philosophy*, 70(17):556–567, 1995.

[31] Xinghan Liu and Emiliano Lorini. A logic for binary classifiers and their explanation. In P. Baroni, C. Benzmüller, and Y. N. Wáng, editors, *Logic and Argumentation - 4th International Conference, CLAR 2021, Hangzhou, China, 2021, Proceedings*, Lecture Notes in Computer Science, pages 302–321. Springer, 2021.

[32] Xinghan Liu and Emiliano Lorini. A logic of "black box" classifier systems. In *Logic, Language, Information, and Computation: 28th International Workshop, WoLLIC 2022, Iași, Romania, 2022, Proceedings*, pages 158–174. Springer Nature, 2022.

[33] Silvan Mertes, Christina Karle, Tobias Huber, Katharina Weitz, Ruben Schlagowski, and Elisabeth André. Alterfactual explanations–the relevance of irrelevance for explaining ai systems. *arXiv preprint arXiv:2207.09374*, 2022.

[34] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[35] Tim Miller. Contrastive explanation: A structural-model approach. *The Knowledge Engineering Review*, 36, 2021.

[36] Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining explanations in AI. In *Proceedings of the 2019 conference on Fairness, Accountability, and Transparency*, pages 279–288, 2019.

[37] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[38] Judea Pearl. *Causality*. Cambridge university press, 2009.

[39] Willard V. Quine. A way to simplify truth functions. *The American mathematical monthly*, 62(9):627–631, 1955.

[40] Weijia Shi, Andy Shih, Adnan Darwiche, and Arthur Choi. On tractable representations of binary neural networks. *arXiv preprint arXiv:2004.02082*, 2020.

[41] Andy Shih, Arthur Choi, and Adnan Darwiche. Formal verification of bayesian network classifiers. In *International Conference on Probabilistic Graphical Models*, pages 427–438. PMLR, 2018.

[42] Kacper Sokol and Peter A. Flach. Counterfactual explanations of machine learning predictions: opportunities and challenges for ai safety. In *SafeAI@ AAAI*, 2019.

[43] Johan Van Benthem, Jan Van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[44] Wiebe van der Hoek, Petar Iliev, and Michael J Wooldridge. A logic of revelation and concealment. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, (AAMAS 2012)*, pages 1115–1122. IFAAMAS, 2012.

[45] Wiebe Van Der Hoek, Nicolas Troquard, and Michael J Wooldridge. Knowledge and control. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, pages 719–726. IFAAMAS, 2011.

[46] Hans van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.

[47] Hans P van Ditmarsch, Wiebe van der Hoek, and Barteld P Kooi. Dynamic epistemic logic with assignment. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 141–148. ACM, 2005.

[48] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[49] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[50] James Woodward. Explanation and invariance in the special sciences. *The British journal for the philosophy of science*, 51(2):197–254, 2000.

[51] James Woodward. *Making Things Happen: a Theory of Causal Explanation*. Oxford University Press, 2003.

[52] James Woodward and Christopher Hitchcock. Explanatory generalizations, part i: A counterfactual account. *Noûs*, 37(1):1–24, 2003.

[53] Fan Yang and Jouko Väänänen. Propositional logics of dependence. *Annals of Pure and Applied Logic*, 167(7):557–589, 2016.