FROM FORMAL CONTRASTIVE TO FORMAL COUNTERFACTUAL EXPLANATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual explanations are often seen as counterfactual conditionals under *closest-world* semantics, requiring one to reason over all minimally distant counterexamples. In contrast, formal XAI typically relies on *feature-based* abductive explanations (AXps) and contrastive explanations (CXps), which admit efficient computation and a minimal hitting-set duality. This paper bridges these perspectives by (i) characterizing closest-world counterfactual explanations explicitly as a disjunctive normal form over all minimally distant counterexample instances, and (ii) introducing *conjunctive counterfactual explanations* (CFXps), a subclass of counterfactual antecedents restricted to conjunctions of feature literals that guarantee prediction change. The paper establishes structural results connecting CXps, AXps and CFXps: every CXp induces an AXp for its witness instance, and every CXp together with a witness AXp induces a CFXp whose support is precisely determined by that AXp. These theoretical results yield a modular approach for computing CFXps using existing AXp/CXp machinery. Experiments on a representative set of benchmarks using tree ensembles and neural networks show that CFXp extraction is practically efficient. Compared with popular point-based counterfactual explainers, CFXps generally require no more feature changes and can guarantee the counterfactual prediction beyond a single instance.

1 INTRODUCTION

Counterfactual reasoning plays a central role in human explanation and decision-making (Miller, 2019). When faced with an outcome, people naturally ask: *"what would need to change for the decision to be different?"*. Recent work (Liu & Lorini, 2023) formalizes this intuition using a modal-logic view of counterfactual conditionals, grounded in closest-world semantics of Stalnaker-Lewis (Stalnaker, 1968; Lewis, 1973). In parallel, formal explainable AI (FXAI) has largely focused on two complementary types of feature-based explanations: abductive explanations (AXps), which justify a prediction, and contrastive explanations (CXps), which identify changes that can alter it (Ignatiev et al., 2020). AXps and CXps enjoy a minimal hitting set duality (Ignatiev et al., 2020), and admit exact computation for a wide range of classifiers (Marques-Silva & Ignatiev, 2022).

Stalnaker-Lewis' counterfactuals (Liu & Lorini, 2023) reason universally over closest counterexamples, whereas CXps require only one (Miller, 2019; Ignatiev et al., 2020). The former allow arbitrary propositional antecedents, while AXps and CXps are restricted to conjunctions of feature-literals. This mismatch creates computational and representational challenges: closest-world counterfactuals can be exponentially (or even infinitely) large and need not correspond to a single feature subset.

We bridge these perspectives by distinguishing two kinds of counterfactual explanations. First, we capture Stalnaker-Lewis' closest-world semantics by a formal closest-world counterfactual explanation: a formula in disjunctive normal form (DNF) over all minimally distant counterexample instances. Second, and more importantly, we introduce an efficiently computable and actionable subclass of formal feature-based counterfactuals, which we call *conjunctive counterfactual explanations*. Like AXps and CXps, they are feature subsets corresponding to conjunctions of feature-literals, but unlike CXps they guarantee, rather than merely enable, a prediction change. We study their properties, relation with AXps and CXps, and computation.

Our main contributions are as follows. First, we formalize Stalnaker-Lewis closest-world counterfactuals in the FXAI setting, building on recent modal-logic work (Liu & Lorini, 2023), and discuss

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

their computation. Second, we introduce *conjunctive* counterfactual explanations (CFXps), a syntactic and computationally tractable form of counterfactual antecedent that blends the existential semantics of CXps with the universal semantics of AXps. Their definition requires neither a distance function nor finite feature domains. Third, we prove two structural results relating CXps, AXps, and CFXps: (1) every CXp induces an AXp for its witness instance, and (2) every CXp induces a CFXp whose support set is precisely characterized by the witness' AXp. Finally, these results yield an efficient and modular algorithm for computing CFXps using the existing FXAI machinery. We evaluate CFXp extraction on XGBoost tree ensembles and ReLU neural networks using independent MaxSAT and MILP backends, and compare it with popular point-based counterfactual explainers. Together, these contributions provide a formal counterfactual reasoning framework and clarify the relations between abductive, contrastive, and conjunctive counterfactual explanations.

2 PRELIMINARIES

2.1 CLASSIFICATION PROBLEMS

Let $\varphi = \{1, \ldots, m\}$, $m \in \mathbb{N}$, be a set of *features*. Subsets of φ will also be denoted by lower-case Greek letters, such as α and β. A feature $j \in \varphi$ is characterized by a set of possible values, called its *domain* $\mathbb{D}_j$. Domain $\mathbb{D}_j$ can be *totally ordered*, in which case feature j is *ordinal* or *numerical*; otherwise, feature j is said to be *categorical*. The *complete feature space* is defined by $\mathbb{F} \triangleq \prod_{j=1}^{m} \mathbb{D}_j$. A *data point* in $\mathbb{F}$ is denoted $\mathbf{v} = (v_1, \ldots, v_m)$, where $v_j \in \mathbb{D}_j$, $j \in \varphi$. Also, we use $\mathbf{x} = (x_1, \ldots, x_m)$ to denote a vector variable in $\mathbb{F}$, where coordinate x_j stores the value of feature j from $\mathbb{D}_j$.

We will also need the notion of *compatible subspaces*. Given a subset of features $\zeta \subseteq \varphi$, define an *equivalence* relation $\sim_\zeta$ on $\mathbb{F}$: $\mathbf{x} \sim_\zeta \mathbf{y}$ if and only if $x_j = y_j, \forall j \in \zeta$. Given a point $\mathbf{v} \in \mathbb{F}$, relation $\sim_\zeta$ defines the equivalence class $[\mathbf{v}]_\zeta \triangleq \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x} \sim_\zeta \mathbf{v}\}$. Moreover, $\sim_\zeta$ splits $\mathbb{F}$ into $|\prod_{j\in\zeta} \mathbb{D}_j|$ equivalence classes, for any $\zeta \in 2^\varphi$. We refer to $[\mathbf{v}]_\zeta$ as the subspace of $\mathbb{F}$ compatible with $\mathbf{v}$ on ζ.

Lemma 1. *For any $\mathbf{v} \in \mathbb{F}$ and feature subsets $\zeta_1 \subseteq \varphi$ and $\zeta_2 \subseteq \varphi$, it holds that*

$$[\mathbf{v}]_{\zeta_1} \cap [\mathbf{v}]_{\zeta_2} \equiv [\mathbf{v}]_{\zeta_1 \cup \zeta_2} \tag{1}$$

Proof (Sketch). By definition, a point $\mathbf{x} \in \mathbb{F}$ belongs to $[\mathbf{v}]_{\zeta_1} \cap [\mathbf{v}]_{\zeta_2}$ iff it agrees with $\mathbf{v}$ on *all* features in both subsets ζ_1 and ζ_2, i.e. on all features in $\zeta_1 \cup \zeta_2$. Hence $[\mathbf{v}]_{\zeta_1} \cap [\mathbf{v}]_{\zeta_2} = [\mathbf{v}]_{\zeta_1 \cup \zeta_2}$. □

Identity (1) is used later. Notably, $[\mathbf{v}]_\zeta$ is simple to represent logically using the conjunction of literals $\bigwedge_{i\in\zeta} (x_i = v_i)$. An assignment of values to variables $x_i \in \mathbb{D}_i$, $i \in \varphi$, trivially satisfies this conjunction iff the data point $\mathbf{x} = (x_1, \ldots, x_m)$ belongs to the compatible subspace, i.e. $\mathbf{x} \in [\mathbf{v}]_\zeta$.

We consider a *total*, *surjective* classification function $\kappa : \mathbb{F} \to \mathbb{K}$, i.e. each point $\mathbf{x} \in \mathbb{F}$ has an associated class $c \in \mathbb{K}$ and, vice versa, each class *"contains"* at least one point mapped to it by κ.[1] A classification problem is characterized by the set of features φ, feature space $\mathbb{F}$, set of classes $\mathbb{K}$, and the classification function κ. Formally, a machine learning (ML) classifier is the tuple $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$. Similar to truth tables in Boolean algebra, classification function κ can in principle be explicitly represented by listing all data points $\mathbf{v} \in \mathbb{F}$ and the corresponding classes $\kappa(\mathbf{v}) = c \in \mathbb{K}$, although depending on the size $|\mathbb{F}|$ this may be *infeasible* or highly inefficient in practice. A row in such a table represented by a pair $\mathbf{i} = (\mathbf{v}, c) \in \mathbb{F} \times \mathbb{K}$ such that $\kappa(\mathbf{v}) = c$ is referred to as an *instance*. Finally, given a classifier M and a concrete instance $\mathbf{i}$, an explanation problem is a tuple $\mathcal{E} = (M, \mathbf{i})$. Classifiers include decision trees (Hyafil & Rivest, 1976), lists (Rivest, 1987) and sets (Clark & Boswell, 1991; Lakkaraju et al., 2016), random forests (Friedman, 2001), boosted trees (Chen & Guestrin, 2016), and neural nets (Nair & Hinton, 2010; Hubara et al., 2016).

Example 1. *Consider classification function $\kappa : \{0,1\}^3 \to \{0,1\}$:*

$$\kappa(x_1, x_2, x_3) = \begin{cases} 1 & \text{if } (x_1 \land x_2) \lor x_3 \\ 0 & \text{otherwise} \end{cases}$$

The function can be fully specified using the truth table shown in Table 1. In the following, we will consider the point $\mathbf{v} = \mathbf{v}_5 = (1, 0, 0)$ classified by κ as 0.

[1] The ideas also extend to regression problems.

Table 1: Truth table for the classification function from Example 1.

	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$\kappa(\mathbf{x})$	0	1	0	1	0	1	1	1

2.2 Interpretability & Formal Explanations

Interpretability is generally accepted to be a subjective concept, without a formal definition (Lipton, 2018). In this paper, we view it in terms of the succinctness of the information used by a classifier to justify a prediction. Building on earlier work, we equate explanations with the so-called *abductive explanations* (AXps) (Shih et al., 2018; Ignatiev et al., 2019b;c; Darwiche & Hirth, 2020; Izza et al., 2020; Ignatiev et al., 2020; Barceló et al., 2020; Marques-Silva et al., 2020; 2021; Huang et al., 2021; Marques-Silva & Ignatiev, 2022; Huang et al., 2022), i.e. subsets of features that are sufficient for the prediction. Formally, given an explanation problem $\mathcal{E} = (M, \mathbf{i})$ with classifier $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$ and instance $\mathbf{i} = (\mathbf{v}, c)$, a *weak AXp* is a subset of features $\alpha \subseteq \varphi$ such that,

$$\forall \mathbf{x} \in [\mathbf{v}]_\alpha. \quad \kappa(\mathbf{x}) = c \tag{2}$$

Weak AXps capture the *selectis paribus* principle ("selected things being equal") (Liu & Lorini, 2023): all points in the subspace of $\mathbb{F}$ compatible with point $\mathbf{v}$ on α are classified as class c. Using the logical formulation of compatible subspaces, (2) can be written as:

$$\forall \mathbf{x} \in \mathbb{F}. \quad \left[\bigwedge_{i \in \alpha} (x_i = v_i)\right] \rightarrow (\kappa(\mathbf{x}) = c) \tag{3}$$

Following Occam's razor, a feature subset $\alpha \subseteq \varphi$ is an abductive explanation (AXp, or strong AXp) if (i) α is a weak AXp and (ii) α is subset-minimal, i.e. no *proper* subset of α is a weak AXp. Some works also refer to AXps as *prime implicant* explanations (or PI-explanations) (Shih et al., 2018) or *sufficient reasons* (Darwiche & Hirth, 2020).

Similarly, a *contrastive explanation* (CXp) (Miller, 2019; Ignatiev et al., 2020) is a feature subset that enables the prediction change when its features take *arbitrary* values from their domains. Formally, a *weak CXp* is a feature subset $\beta \subseteq \varphi$ such that,

$$\exists \mathbf{x} \in [\mathbf{v}]_{\varphi \setminus \beta}. \quad \kappa(\mathbf{x}) \neq c \tag{4}$$

Unlike AXps, weak CXps capture the *ceteris paribus* principle ("all other things being equal"): there is at least one point in the subspace of $\mathbb{F}$ compatible with $\mathbf{v}$ on $\varphi \setminus \beta$ that is classified differently from class c. Using the logical formulation of compatible subspaces, (4) can be written as:

$$\exists \mathbf{x} \in \mathbb{F}. \quad \left[\bigwedge_{i \in \varphi \setminus \beta} (x_i = v_i)\right] \land (\kappa(\mathbf{x}) \neq c) \tag{5}$$

A feature subset $\beta \subseteq \varphi$ is a contrastive explanation (CXp, or strong CXp) if it is a subset-minimal weak CXp. CXps are also referred to as *necessary reasons* (Darwiche & Ji, 2022).

Recent work showed that AXps and CXps satisfy a minimal hitting set (MHS) duality (Ignatiev et al., 2020): each AXp is a minimal hitting set of the set of CXps, and vice versa. Given an explanation problem $\mathcal{E} = (M, \mathbf{i})$ characterized by a classifier M and a particular instance $\mathbf{i} = (\mathbf{v}, c)$, let $\mathbb{A}_\mathcal{E}$ denote the complete set of AXps and $\mathbb{B}_\mathcal{E}$ denote the complete set of CXps. Then $\mathbb{A}_\mathcal{E} = \text{MHS}(\mathbb{B}_\mathcal{E})$ and $\mathbb{B}_\mathcal{E} = \text{MHS}(\mathbb{A}_\mathcal{E})$, where $\text{MHS}(S)$ returns all minimal hitting sets of S, i.e. minimal sets that share an element with each subset in S. Formally, $\text{HS}(S) = \{t \subseteq (\cup S) \mid \forall s \in S,\ t \cap s \neq \emptyset\}$, $\text{mins}(S) = \{s \in S \mid \forall t \subsetneq s,\ t \notin S\}$ returns the subset-minimal elements of a set of sets, and $\text{MHS}(S) = \text{mins}(\text{HS}(S))$. This MHS duality has enabled efficient algorithms for computing and enumerating AXps and CXps (Ignatiev et al., 2020; Marques-Silva & Ignatiev, 2022).

Example 2. *Consider the explanation problem $\mathcal{E} = (M, \mathbf{i})$, with classifier M characterized by the features $\varphi = \{1, 2, 3\}$, feature space $\mathbb{F} = \{0, 1\}^3$, classes $\mathbb{K} = \{0, 1\}$, as well as classification function $\kappa(x_1, x_2, x_3)$ and the instance $\mathbf{i} = (\mathbf{v}_5, 0)$, $\mathbf{v}_5 = (1, 0, 0)$ from Example 1. By inspection of Table 1, the only AXp is $\alpha = \{2, 3\}$, since fixing $x_2 = 0$ and $x_3 = 0$ guarantees class 0. The two CXps are $\beta_1 = \{2\}$ and $\beta_2 = \{3\}$, witnessed by points $\mathbf{v}_7 = (1, 1, 0)$ and $\mathbf{v}_6 = (1, 0, 1)$, respectively. Hence $\mathbb{A}_{\mathcal{E}} = \{\{2, 3\}\}$ and $\mathbb{B}_{\mathcal{E}} = \{\{2\}, \{3\}\}$, illustrating the hitting-set duality.*

Related explanation problems have been studied for compiled representations and Boolean classifier families (Audemard et al., 2020; 2021; Darwiche & Marquis, 2021). Other extensions incorporate background knowledge in the form of domain constraints (Gorji & Rubin, 2022; Cooper & Amgoud, 2023), including knowledge extracted from data (Yu et al., 2023), or generalize AXps by associating them with Craig interpolants (Craig, 1957; McMillan, 2003; Labbaf et al., 2025). Formal guarantees further cover robust NLP explanations (Malfa et al., 2021), approximate formal neural explanations (Bassan & Katz, 2023), and counterfactual anomaly repair (Ji et al., 2024). Complementary axiomatic and complexity-theoretic accounts study general properties of counterfactual and formal explanations (Amgoud & Cooper, 2026; Bassan et al., 2026).

3 Formal Closest-World Counterfactuals

This work builds on a recent modal-logic account (Liu & Lorini, 2023), which uses epistemic *modal logic* to define abductive and contrastive explanations via *necessity* ($\Box$) and *possibility* ($\Diamond$). It also relates counterfactual explanations for ML models (Miller, 2019; Molnar, 2020) to *counterfactual conditionals* under closest-world semantics of Stalnaker-Lewis (Stalnaker, 1968; Lewis, 1973).

Informally, a counterfactual conditional $\tau \Rightarrow \rho$ holds *"if all closest worlds to the actual world in which the antecedent τ is true satisfy the consequent ρ"*. For an explanation problem $\mathcal{E} = (M, \mathbf{i})$, the *worlds* represent data points $\mathbf{x} \in \mathbb{F}$ closest to the point of interest $\mathbf{v} \in \mathbb{F}$, which represents the actual world. Following (Liu & Lorini, 2023, Proposition 3), closest worlds can be constructed using the Hamming distance between $\mathbf{x}$ and $\mathbf{v}$. For two points $\mathbf{x}, \mathbf{y} \in \mathbb{F}$, their *Hamming distance* is the number of coordinates on which they disagree, i.e.

$$d_H(\mathbf{x}, \mathbf{y}) \triangleq |\{i \in \varphi \mid x_i \neq y_i\}|.$$

Here, the antecedent τ may be an arbitrary Boolean formula serving as a counterfactual explanation for $\mathcal{E} = (M, \mathbf{i})$, $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$ and $\mathbf{i} = (\mathbf{v}, c)$, while the consequent ρ of the conditional is meant to be the predicate $\kappa_{\bar{c}} \triangleq (\kappa(\mathbf{x}) \neq c)$. One natural choice for τ is a disjunctive normal form (DNF) whose terms represent the closest points to $\mathbf{v}$ where $\kappa_{\bar{c}}$ holds. Each term need only include literals on the features that *differentiate* the corresponding closest point from the point of interest $\mathbf{v}$.

Example 3. *For the point $\mathbf{v} = \mathbf{v}_5 = (1, 0, 0)$, the Hamming-closest points are $\mathbf{v}_1 = (0, 0, 0)$, $\mathbf{v}_6 = (1, 0, 1)$, and $\mathbf{v}_7 = (1, 1, 0)$. Only $\mathbf{v}_7$ and $\mathbf{v}_6$ change the prediction, so the closest-world antecedent is $(x_2 \vee x_3)$, yielding $(x_2 \vee x_3) \Rightarrow (\kappa(\mathbf{x}) = 1)$.*

Closest-world counterfactual explanations thus capture universal semantics akin to abductive explanations, i.e. *if a formula holds on all closest points, the class is guaranteed to differ*. By contrast, contrastive explanations require only a witness for prediction change. While (Liu & Lorini, 2023) gives no algorithm for these explanations or their closest points, we give one below.

For an explanation problem $\mathcal{E} = (M, \mathbf{i})$ with $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$ and $\mathbf{i} = (\mathbf{v}, c)$, (Liu & Lorini, 2023) considers the points $\mathbf{x} \in \mathbb{F}$ with the *smallest* Hamming distance from $\mathbf{v}$ such that $\kappa(\mathbf{x}) \neq c$. Let $d_H^{\min}(\mathbf{v}) = \min\{d_H(\mathbf{v}, \mathbf{x}) \mid \mathbf{x} \in \mathbb{F}, \kappa(\mathbf{x}) \neq c\}$. Then the set of Hamming-closest points to be used in the closest-world counterfactual argument for $\mathbf{v}$ can be expressed as

$$\mathbb{E}_{\mathbf{v}} \triangleq \{\mathbf{x} \in \mathbb{F} \mid d_H(\mathbf{v}, \mathbf{x}) = d_H^{\min}(\mathbf{v}) \text{ and } \kappa(\mathbf{x}) \neq c\}.$$

The set $\mathbb{E}_{\mathbf{v}}$ defines the DNF formula

$$\tau(\mathbf{x}) \triangleq \bigvee_{\mathbf{u} \in \mathbb{E}_{\mathbf{v}}} \left[\bigwedge_{j \in \varphi} (x_j = u_j) \right] \tag{6}$$

serving as a counterfactual explanation for $\mathbf{v} \in \mathbb{F}$. Since τ comprises all minimally distant points predicted differently from $\mathbf{v}$, we call (6) a *closest-world* or *disjunctive* counterfactual explanation.

Computing a closest-world counterfactual explanation $\mathbb{E}_{\mathbf{v}}$ requires enumerating all *cardinality-minimal* CXps $\beta \subseteq \varphi$ and all their witnesses $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\beta}$ such that $\kappa(\mathbf{y}) \neq c$. This exhaustive process is expensive and may be infeasible when $\mathbb{F}$ is infinite and lacks a succinct interval representation. Additionally, if $\mathbb{E}_{\mathbf{v}}$ is finite, one may further seek a minimum-size DNF equivalent to τ with an algorithm similar to the Quine-McCluskey (Quine, 1952; 1955; McCluskey, 1956) procedure, thus solving the DNF minimization problem, which is Σ_2^P-hard (Umans, 1998; Umans et al., 2006).

Example 4. *For $\mathcal{E} = (M, (\mathbf{v}_5, 0))$, the smallest CXps are $\beta_1 = \{2\}$ and $\beta_2 = \{3\}$, witnessed by $\mathbf{v}_7$ and $\mathbf{v}_6$, respectively. Here $\tau(\mathbf{x}) = (x_1 \land x_2 \land \neg x_3) \lor (x_1 \land \neg x_2 \land x_3)$. Dropping the literals shared with $\mathbf{v}_5$ reduces it to $x_2 \lor x_3$ (Stalnaker, 1968; Lewis, 1973; Liu & Lorini, 2023).*

4 Formal Conjunctive Counterfactuals

The closest-world counterfactual explanations above capture the exact semantics of Stalnaker-Lewis' counterfactuals but pose computational and interpretability challenges. DNF minimization may compress an extracted explanation but cannot reduce the cost of closest-world enumeration. We therefore consider a less general explanation that still captures the counterfactual argument and guarantees prediction change. We call it a *conjunctive counterfactual explanation*, restricted to conjunctions of literals rather than arbitrary formulas defined via minimal Hamming distance.[2]

Such counterfactuals blend AXps and CXps: like CXps, they concern prediction change, but CFXps *guarantee*, i.e. logically entail, it. We use notation similar to that of abductive and contrastive explanations, where an explanation need not be (cardinality- or) subset-minimal.

Definition 1. *Let $\mathcal{E} = (M, \mathbf{i})$ be an explanation problem with classifier $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$ and instance $\mathbf{i} = (\mathbf{v}, c)$. A weak* (conjunctive) counterfactual explanation *(weak CFXp) is a subset of features $\gamma \subseteq \varphi$ such that,*

$$\exists \delta \subseteq \varphi \setminus \gamma, \exists \mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\gamma}, \forall \mathbf{x} \in [\mathbf{y}]_\gamma \cap [\mathbf{v}]_\delta. \quad \kappa(\mathbf{x}) \neq c \tag{7}$$

Point $\mathbf{y}$ is a *witness* for γ: it is *close enough* to $\mathbf{v}$ but predicted differently, and thus determines the minimal changes from $\mathbf{v}$. Support δ fixes values shared by $\mathbf{y}$ and $\mathbf{v}$ so that every point in $[\mathbf{y}]_\gamma \cap [\mathbf{v}]_\delta$ has a prediction other than c, i.e. it *"pins down what stays fixed"*. Thus, δ supports γ and helps minimize it, capturing the counterfactual conditional: γ controls *"variance"* from $\mathbf{v}$, while δ controls *"invariance"*. We write $(\gamma, \mathbf{y}, \delta)$ for a CFXp with its witness and support. Features in γ take their values from $\mathbf{y}$, features in δ retain their values from $\mathbf{v}$, and all others may vary while the counterfactual prediction remains guaranteed. (The case $\delta = \varphi \setminus \gamma$ is degenerate because $[\mathbf{y}]_\gamma \cap [\mathbf{v}]_{\varphi\setminus\gamma} = \{\mathbf{y}\}$, making the explanation *too narrow*; $\delta = \emptyset$ may also be degenerate when points $[\mathbf{y}]_\gamma \cap [\mathbf{v}]_\emptyset$ are *too far* from $\mathbf{v}$. We show below that a "good" δ can be computed efficiently.)

As with AXps and CXps, a (strong conjunctive) counterfactual explanation (CFXp) is a subset-minimal weak CFXp $\gamma \subseteq \varphi$. For CFXps, targeting a specific prediction may be more useful than requiring *any other* prediction. For $q \in \mathbb{K}$, $q \neq c$, a weak *targeted counterfactual explanation* (weak targeted CFXp) is a subset γ such that,

$$\exists \delta \subseteq \varphi \setminus \gamma, \exists \mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\gamma}, \forall \mathbf{x} \in [\mathbf{y}]_\gamma \cap [\mathbf{v}]_\delta. \quad \kappa(\mathbf{x}) = q \tag{8}$$

This specifies which features change, and how, to reach class $q \neq c$.

Example 5. *For $\mathbf{v} = (1, 0, 0)$, one CFXp is $\gamma_1 = \{2\}$, witnessed by $\mathbf{v}_7 = (1, 1, 0)$ and supported by $\delta_1 = \{1\}$, since every point in $[\mathbf{v}_7]_{\{2\}} \cap [\mathbf{v}]_{\{1\}}$ is classified as 1. Thus, the triple $(\gamma_1, \mathbf{v}_7, \delta_1) = (\{2\}, (1, 1, 0), \{1\})$ gives the counterfactual argument:* "changing x_2 to 1 while keeping x_1 fixed at 1 will lead to prediction 1, regardless of the value of x_3". *Another is $\gamma_2 = \{3\}$, witnessed by $\mathbf{v}_6 = (1, 0, 1)$ with support $\delta_2 = \emptyset$, since every point in $[\mathbf{v}_6]_{\{3\}}$ is classified as 1. The triple $(\gamma_2, \mathbf{v}_6, \delta_2) \triangleq (\{3\}, (1, 0, 1), \emptyset)$ gives the counterfactual argument:* "changing x_3 to 1 regardless of the values of x_1 and x_2 leads to prediction 1".

Both contrastive and counterfactual explanations suggest how to change the prediction and are often called *actionable* (Karimi et al., 2023; Singh et al., 2024). Whereas CXps guarantee one nearby

[2] Conjunctive counterfactual explanations are not semantically equivalent to the closest-world explanations of the modal-logic account (Liu & Lorini, 2023), but form a syntactic subset of counterfactual antecedents designed for better scalability and interpretability.

instance predicted differently from $\mathbf{v}$, CFXps may offer a family of differently predicted instances based on γ. In this sense, CFXps can be deemed stronger than CXps and viewed as their generalization, as the following proposition shows:

Proposition 1. *Let $\mathcal{E} = (M, \mathbf{i})$ be an explanation problem with $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$ and $\mathbf{i} = (\mathbf{v}, c) \in \mathbb{F} \times \mathbb{K}$. A subset $\gamma \subseteq \varphi$ is a weak targeted (untargeted, resp.) CFXp if and only if it is a weak targeted (untargeted, resp.) CXp. As a result, the subset-minimal feature sets of targeted (untargeted, resp.) CFXps and targeted (untargeted, resp.) CXps coincide.*

Proof. Let γ be a weak *untargeted* CFXp witnessed by point $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\gamma}$ and supported by $\delta \subseteq \varphi\setminus\gamma$. Since $\mathbf{y} \in [\mathbf{y}]_\gamma \cap [\mathbf{v}]_\delta$, (7) implies $\kappa(\mathbf{y}) \neq c$. Hence, γ is a weak CXp witnessed by $\mathbf{y}$. Conversely, let γ be a weak CXp witnessed by a point $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\gamma}$ such that $\kappa(\mathbf{y}) \neq c$. Pick $\delta = \varphi \setminus \gamma$ and observe that $[\mathbf{y}]_\gamma \cap [\mathbf{v}]_\delta = \{\mathbf{y}\}$. Hence, (7) holds and γ is a weak CFXp. The targeted case follows similarly by using (8) and the definition of targeted CXps (Ignatiev et al., 2020). Thus, the feature sets of weak CFXps and weak CXps coincide, as do their subset-minimal feature sets. □

Proposition 1 shows that CFXps and CXps have the same subset-minimal feature sets. A CFXp also associates γ with a witness $\mathbf{y}$ and support δ, guaranteeing the counterfactual prediction beyond $\mathbf{y}$.

Because they capture entailment, *untargeted* CFXps relate to the negation of prime *implicates* of the predicate $\kappa_c \triangleq (\kappa(\mathbf{v}) = c)$. Similarly, *targeted* CFXps relate to *absolute* (global or class) abductive explanations (Ignatiev et al., 2019c) (see also Section 5). Given an instance $\mathbf{i} = (\mathbf{v}, c)$, a CFXp γ witnessed by $(\mathbf{y}, q)$ and supported by δ guarantees the prediction $q \neq c$. Equivalently, $\gamma \cup \delta$ applied to $\mathbf{y}$ is either a counterexample for class $c \in \mathbb{K}$ or an absolute explanation for class $q \neq c$. Next, we show how to compute targeted CFXps and relate them to AXps and CXps.

5 Computing Conjunctive Counterfactuals

We focus solely on *targeted* CFXps, omitting "targeted" for brevity. We also focus on strong (subset-minimal) explanations. Given an instance of interest $\mathbf{i} = (\mathbf{v}, c) \in \mathbb{F} \times \mathbb{K}$, we show how a CXp $\beta \subseteq \varphi$ for $\mathbf{i}$ can be used to construct an AXp $\alpha \subseteq \varphi$ for another instance $(\mathbf{y}, q)$, $q \neq c$, witnessing β.

Proposition 2. *Let $\beta \subseteq \varphi$ be a CXp for instance $\mathbf{i} = (\mathbf{v}, c)$. Also, let $(\mathbf{y}, q)$ be an arbitrary witness of β, i.e. $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\beta}$. Then there is no AXp α for instance $(\mathbf{y}, q)$ such that $\alpha \subsetneq \beta$.*

Proof. Let β be a CXp for $\mathbf{i} = (\mathbf{v}, c)$ with witness $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\beta}$ satisfying $\kappa(\mathbf{y}) = q$. Assume, by contradiction, that instance $(\mathbf{y}, q)$ has an AXp $\alpha \subsetneq \beta$. By (2), $\forall \mathbf{x} \in [\mathbf{y}]_\alpha.\ \kappa(\mathbf{x}) = q$. Construct a point $\mathbf{u} \in [\mathbf{y}]_\alpha \cap [\mathbf{v}]_{\varphi\setminus\alpha}$ by setting $u_i = y_i$ for $i \in \alpha$ and $u_j = v_j$ for $j \in \varphi \setminus \alpha$. Then $\mathbf{u} \in [\mathbf{y}]_\alpha$, and so $\kappa(\mathbf{u}) = q$. Yet $\mathbf{u}$ differs from $\mathbf{v}$ only on features $\alpha \subsetneq \beta$, contradicting that β is a CXp. □

Proposition 3. *Let $\beta \subseteq \varphi$ be a CXp for instance $\mathbf{i} = (\mathbf{v}, c)$ with an arbitrary witness $(\mathbf{y}, q)$, where $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\beta}$ and $\kappa(\mathbf{y}) = q \neq c$. Then there is an AXp α for instance $(\mathbf{y}, q)$ such that $\alpha \supseteq \beta$.*

Proof. Let $\beta \subseteq \varphi$ be a CXp for instance $\mathbf{i} = (\mathbf{v}, c)$ and fix a witness $\mathbf{y} \in [\mathbf{v}]_{\varphi\setminus\beta}$ with $\kappa(\mathbf{y}) = q$. Consider the following set of feature subsets:

$$Z = \{\zeta \subseteq \varphi \mid \beta \subseteq \zeta, \forall \mathbf{x} \in [\mathbf{y}]_\zeta.\ \kappa(\mathbf{x}) = q\}.$$

Thus, Z is the set of all weak AXps of $(\mathbf{y}, q)$ that include β. It is non-empty because $\varphi \in Z$, as $[\mathbf{y}]_\varphi = \{\mathbf{y}\}$. Since Z is finite, it has a minimal element $\alpha \in Z$ wrt. *set inclusion*.

By construction, α satisfies three conditions: (1) $\beta \subseteq \alpha$, (2) $\forall \mathbf{x} \in [\mathbf{y}]_\alpha.\ \kappa(\mathbf{x}) = q$, i.e. α is a weak AXp for $(\mathbf{y}, q)$, and (3) minimality within Z.

It remains to show that α is *globally* subset-minimal, not only among weak AXps containing β. Suppose, for contradiction, that there exists $\alpha' \subsetneq \alpha$ such that $\forall \mathbf{x} \in [\mathbf{y}]_{\alpha'}.\ \kappa(\mathbf{x}) = q$. Then necessarily $\alpha' \not\supseteq \beta$; otherwise, the minimality of α in Z would be contradicted.

Construct a point $\mathbf{u}$ by fixing all features in α' to $\mathbf{y}$'s values and all other features to $\mathbf{v}$'s values:

$$u_j = \begin{cases} y_j, & j \in \alpha' \\ v_j, & j \notin \alpha' \end{cases} \quad \Longrightarrow \quad \mathbf{u} \in [\mathbf{y}]_{\alpha'} \cap [\mathbf{v}]_{\varphi\setminus\alpha'}.$$

Since α' is, by assumption, a weak AXp for $(\mathbf{y}, q)$ and $\mathbf{u} \in [\mathbf{y}]_{\alpha'}$, we have $\kappa(\mathbf{u}) = q$.

Recall that $\mathbf{y}$ is a witness for the CXp β, so we have $\{j \in \varphi \mid y_j \neq v_j\} = \beta$. Given that $u_j = v_j$ for all $j \notin \alpha'$, the coordinates on which $\mathbf{u}$ and $\mathbf{v}$ *actually* differ are exactly:

$$\{j \in \varphi \mid u_j \neq v_j\} = \{j \in \alpha' \mid y_j \neq v_j\} = \alpha' \cap \beta \subseteq \beta.$$

Since $\alpha' \not\supseteq \beta$, there is a feature $i \in \beta \setminus \alpha'$, and so $\alpha' \cap \beta \subsetneq \beta$. Thus, we have produced a point $\mathbf{u}$ with $\kappa(\mathbf{u}) = q$ that differs from $\mathbf{v}$ only on features $\alpha' \cap \beta \subsetneq \beta$. This contradicts that β is a CXp. □

Proposition 3 has two important consequences. First, it yields an efficient way to compute an AXp for an instance $(\mathbf{y}, q)$ once this instance is known to witness a CXp β for $(\mathbf{v}, c)$. The AXp computation can bypass traversing the entire feature set φ, focusing only on features $\varphi \setminus \beta$. This is useful when multiple instances from different classes are explained in sequence, both in terms of AXps and CXps. It may help when compiling an ML model into a decision set or enumerating explanations (Ignatiev et al., 2020) for multiple instances.

Second, Proposition 3 provides the basis for computing CFXps:

Proposition 4. *Let $\mathcal{E} = (M, \mathbf{i})$ be an explanation problem with $M = (\varphi, \mathbb{F}, \mathbb{K}, \kappa)$ and $\mathbf{i} = (\mathbf{v}, c) \in \mathbb{F} \times \mathbb{K}$. Let $\beta \subseteq \varphi$ be a CXp for $\mathbf{i}$ witnessed by $(\mathbf{y}, q)$, and let $\alpha \supseteq \beta$ be an AXp for $(\mathbf{y}, q)$. Then β is a CFXp for $\mathbf{i}$, witnessed by $\mathbf{y}$ and supported by $\delta = \alpha \setminus \beta$.*

Proof. We first show that β is a weak CFXp satisfying (8). Since α is an AXp for $(\mathbf{y}, q)$, $\forall \mathbf{x} \in [\mathbf{y}]_\alpha$. $\kappa(\mathbf{x}) = q$. Since $\mathbf{y}$ witnesses β, it agrees with $\mathbf{v}$ on $\varphi \setminus \beta$, and hence on $\delta = \alpha \setminus \beta$. Thus, $[\mathbf{v}]_\delta = [\mathbf{y}]_\delta$. Hence, $[\mathbf{y}]_\beta \cap [\mathbf{v}]_\delta = [\mathbf{y}]_\beta \cap [\mathbf{y}]_\delta$. By Lemma 1, $[\mathbf{y}]_\beta \cap [\mathbf{y}]_\delta = [\mathbf{y}]_{\beta \cup \delta} = [\mathbf{y}]_\alpha$. Therefore, $\forall \mathbf{x} \in [\mathbf{y}]_\beta \cap [\mathbf{v}]_\delta$. $\kappa(\mathbf{x}) = q$, so (8) holds and β is a weak CFXp.

It remains to show that β is subset-minimal. Otherwise, some $\beta' \subsetneq \beta$ is a weak CFXp. Then, by Proposition 1, β' is a weak CXp, contradicting that β is a CXp. Hence, β is a CFXp. □

By Proposition 4, any CXp for $\mathbf{i} = (\mathbf{v}, c)$ can be declared a CFXp by computing the witness and support as follows:

1. find any CXp β for instance $\mathbf{i}$ with witness $(\mathbf{y}, q)$;
2. compute any AXp $\alpha \supseteq \beta$ for $(\mathbf{y}, q)$;
3. output a triple $(\beta, \mathbf{y}, \alpha \setminus \beta)$.

The triple $(\beta, \mathbf{y}, \alpha \setminus \beta)$ gives the CFXp, its witness, and its support, respectively. The support need not be unique: different traversal orders may yield different subset-minimal supports without affecting correctness.

This construction neither uses a distance function nor enumerates closest counterexamples: it computes a CXp and witness followed by an AXp for that witness. With deletion-based extraction, both steps make linearly many reasoner calls; once β and its witness are known, the second tests at most $|\varphi| - |\beta|$ features. These are call bounds only; each call's cost depends on the model representation and reasoning engine. Closest-world extraction is also output-sensitive because the number of closest counterexamples may be exponential.

Example 6. *In the running example, $\gamma_1 = \{2\}$ and $\gamma_2 = \{3\}$ are obtained from CXps $\beta_1 = \{2\}$ and $\beta_2 = \{3\}$ by computing AXps for witnesses $\mathbf{v}_7$ and $\mathbf{v}_6$. From $\mathbf{v}_7$ we obtain $\alpha_1 = \{1, 2\}$ and the triple $(\gamma_1, \mathbf{v}_7, \{1\})$ by testing features 1 and 3 but, by Proposition 3, not feature $2 \in \beta_1$. For β_2, we analogously obtain $(\gamma_2, \mathbf{v}_6, \emptyset)$.*

The *AXp-CXp-CFXp triad* holds only for conjunctive CFXps, not for closest-world counterfactuals (Liu & Lorini, 2023): CFXps *relax* the exhaustive universal condition on minimal-distance instances (Stalnaker, 1968; Lewis, 1973) into the more actionable existential–universal pattern (8).

6 Experimental Evidence

This section presents experimental results assessing the performance of a single CFXp extraction on widely studied public datasets. First, we compare CFXps with closest-world counterfactual explanations (CWCs), which directly instantiate the closest-world semantics formalized in Section 4.

Second, we compare CFXps with well-established point-based counterfactual explainers on XGBoost tree ensembles and ReLU neural networks.

Experimental setup. The experiments were conducted on a MacBook Pro with an Apple M1 Pro CPU with 10 cores and 32 GByte RAM running macOS 26.2 Tahoe. The time limit for extracting an individual formal explanation is set to 5 minutes; no memory limit is imposed. We use 21 tabular datasets from the UCI Machine Learning Repository (Kelly et al., 2023) and Penn Machine Learning Benchmarks (PMLB) (Olson et al., 2017; Romano et al., 2021). The number of original features ranges from 7 to 60 and the number of instances from 59 to 58000, with averages of 26.95 and 4684.47, respectively. For each dataset, we use the same randomly selected set of at most 200 instances, resulting in 3755 explained predictions.

Implementations. For tree ensembles, we reuse the XGBoost models and experimental setup of the earlier MaxSAT study (Ignatiev et al., 2022). The models contain 50 trees per class of depth at most 3–4, and their propositional encodings use 4–10183 variables and 304–102611 clauses. The MaxSAT approach (Ignatiev et al., 2022) encodes tree-induced intervals and paths as hard clauses and retains floating-point leaf scores as weights of soft objective literals; feature relevance is tested by incrementally invoking the RC2 MaxSAT solver (Ignatiev et al., 2018; 2019a). For neural networks, we train a fully connected ReLU model on each of the same 21 datasets, with hidden layers of 16 and 8 neurons. The resulting models have 7–60 inputs, 2–11 classes, and 273–1121 trainable parameters. They are encoded into MILP using a standard ReLU encoding (Fischetti & Jo, 2018) and solved with Gurobi (Gurobi Optimization, 2026). Thus, the experiments use two independent reasoning backends for CXp, AXp, and CFXp extraction.

Comparison with closest-world explanations. This experiment is restricted to tree ensembles, whose thresholds partition every continuous feature domain into a finite number of non-overlapping intervals. As a result, closest worlds can be enumerated over the corresponding regions. To this end, we extend the MaxSAT approach for tree ensembles (Ignatiev et al., 2022) to enumerate all smallest-size CXps and all their witnesses. In the following, this procedure is referred to as CWC and is compared with the computation of one AXp, CXp, and CFXp per prediction.

CFXp computation remains comparable to AXp and CXp computation, as it consists of computing a CXp followed by an AXp. All three methods finish on all 3755 instances, with mean/median runtimes of 1.529/0.266, 0.783/0.184, and 2.098/0.500 seconds for AXp, CXp, and CFXp extraction, respectively. By contrast, CWC reaches the 300-second time limit in 428 cases. Keeping these timeout observations capped at 300 seconds gives a mean/median runtime of 51.872/2.168 seconds. The true mean is necessarily higher because these 428 runs did not finish within the time limit.

The higher computational cost of CWC should be attributed to the number of closest worlds that must be enumerated before the computation can finish. For the 3327 completed cases, this number ranges from 1 to 17697, with mean/median 117.770/19. Regarding explanation size, AXps have mean/median size 9.043/8, while the CFXp feature set γ, its support δ, and the complete certificate have mean/median sizes 1.972/1, 8.240/7, and 10.212/8, respectively. Although the number of worlds and the number of features measure different objects, these are nevertheless the natural sizes of the two representations. The complete runtime and size distributions, including cactus and per-instance scatter plots, are provided in the appendix.

Comparison with point-based explainers. We next compare CFXps with DiCE (Mothilal et al., 2020), Growing Spheres (Laugel et al., 2017), and the optimization-based method of Wachter et al. (Wachter et al., 2018). Each method is asked for one point whose prediction differs from that of the explained instance. All returned points are checked independently against the corresponding model; hence, the success counts in Table 2 include only model-verified points. For multiclass problems, CFXps and Growing Spheres are untargeted, whereas DiCE and Wachter require a target class. Therefore, we run the latter for every class other than the original prediction and select the point changing the fewest features. All three methods are evaluated on the same 3755 predictions from all 21 datasets.

As shown in Table 2, DiCE, Wachter, and Growing Spheres return a model-verified result in 99.97–100%, 89.40–92.17%, and 98.3–100% of all cases. CFXp extraction finishes on all 3755 predictions for both model classes. For the neural models, $|\gamma|$ ranges from 1 to 25 with mean/median 2.064/1,

Table 2: Comparison with point-based counterfactual explainers. Size and time entries report *mean/median*. L_0 distance denotes the number of features whose values differ from the original instance. These and paired CFXp $|\gamma|$ are computed over model-verified point outputs; CFXp$\leq$ reports the fraction of these pairs in which $|\gamma|$ is no larger. Times cover all attempted instances and use CPU time for XGBoost and wall time for ReLU networks.

Model	Point method	Verified	L_0 distance	CFXp $\|\gamma\|$	CFXp$\leq$	Point time (s)	CFXp time (s)
XGBoost	DiCE	3754/3755	2.310/ 2	1.971/1	87.24%	0.483/ 0.208	2.098/0.500
	Growing Spheres	3755/3755	3.512/ 3	1.972/1	95.31%	5.518/ 3.428	2.098/0.500
	Wachter	3461/3755	20.588/16	1.969/1	100.00%	41.128/23.985	2.098/0.500
ReLU	DiCE	3755/3755	2.083/ 2	2.064/1	79.92%	0.408/ 0.196	0.310/0.076
	Growing Spheres	3692/3755	9.836/ 9	1.882/1	97.24%	27.550/11.325	0.310/0.076
	Wachter	3357/3755	20.120/15	1.982/1	99.73%	4.163/ 2.588	0.310/0.076

while extraction takes from 0.005 to 27.749 seconds with mean/median 0.310/0.076 seconds. The support δ and complete *certificate* $\gamma \cup \delta$ have mean/median sizes 20.577/17 and 22.641/19, respectively. For context, ordinary AXps for these models have mean/median size 19.734/15, which helps explain the larger support sets. Importantly, these support features remain fixed rather than being changed. Furthermore, for neural models the CFXp changed-feature set γ is no larger than the number of changed features in the explanation of DiCE, Wachter, and Growing Spheres, in 79.92%, 99.73%, and 97.24% of the paired cases. For XGBoost models, the CFXp changed-feature set γ is no larger in 87.24%, 100%, and 95.31% of the paired cases. These results should not be interpreted as a general ranking of the methods: in particular, the Wachter method optimizes normalized L_1 distance instead of L_0, while DiCE is faster than the MaxSAT-based CFXp implementation on XGBoost. Nevertheless, the results show that the strong regional guarantee provided by a CFXp generally does not require more changed features than a single model-verified point.

Summary. The experiments show that CFXp extraction is practically efficient for tree ensembles and ReLU neural networks using two independent reasoning backends. Unlike CWC, CFXps require neither Hamming distance nor exhaustive closest-world enumeration, and are not restricted to finite feature domains. Compared with popular point-based explainers, CFXps generally require no more feature changes while guaranteeing the counterfactual prediction beyond one returned instance.

7 Conclusions

This work has developed a formal framework for counterfactual explanations in classifier systems that connects closest-world counterfactuals (Stalnaker, 1968; Lewis, 1973; Liu & Lorini, 2023) with feature-based explanations used in formal XAI (Marques-Silva & Ignatiev, 2022). We instantiated the modal-logic framework of (Liu & Lorini, 2023) in a feature-space setting, obtaining disjunctive closest-world counterfactual explanations under the Hamming-style similarity measure. We then introduced conjunctive counterfactual explanations (CFXps), a restricted yet actionable class of formal feature-based counterfactuals that guarantee prediction change and align naturally with abductive explanations (AXps) and contrastive explanations (CXps). Unlike closest-world counterfactuals, CFXps require no distance function and are not restricted to finite feature domains.

First, we established a triad linking AXps, CXps, and CFXps. Feature-wise, every CFXp is a CXp, and every CXp together with an AXp for its witness instance induces a CFXp whose support set is characterized by that AXp, yielding a modular algorithm that reuses existing AXp/CXp machinery. Second, our experiments on XGBoost tree ensembles and ReLU neural networks show that CFXp extraction is practically efficient. For tree ensembles, CFXp computation remains comparable to single AXps or CXps, while closest-world counterfactual explanations require costly exhaustive enumeration. Compared with popular point-based explainers, CFXps generally require no more feature changes while providing a guarantee beyond one returned point.

Overall, CFXps offer a viable compromise between the full closest-world semantics of modal counterfactuals and the combinatorial structure of formal AXps and CXps, providing sound, efficiently computable, and comparatively succinct counterfactual arguments within the formal XAI toolbox.

REPRODUCIBILITY STATEMENT

Appendix A details the benchmarks, models, solver and baseline configurations, and complete experimental results. The supplementary material includes all 21 datasets, trained models, source code, dependency versions, scripts for running the experiments and reproducing our experimental results, and raw results. The source code, datasets, and the entire experimental setup will be publicly released upon acceptance of this submission.

REFERENCES

Leila Amgoud and Martin C. Cooper. Axiomatic foundations of counterfactual explanations. In *AAMAS*, pp. 282–290, 2026.

Gilles Audemard, Frédéric Koriche, and Pierre Marquis. On tractable XAI queries based on compiled representations. In *KR*, pp. 838–849, 2020.

Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the computational intelligibility of boolean classifiers. *CoRR*, abs/2104.06172, 2021. URL https://arxiv.org/abs/2104.06172.

Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. Model interpretability through the lens of computational complexity. In *NeurIPS*, 2020.

Shahaf Bassan and Guy Katz. Towards formal XAI: formally approximate minimal explanations of neural networks. In *TACAS (1)*, pp. 187–207, 2023.

Shahaf Bassan, Xuanxiang Huang, and Guy Katz. Unifying formal explanations: A complexity-theoretic perspective. *CoRR*, abs/2602.18160, 2026.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *KDD*, pp. 785–794, 2016.

Peter Clark and Robin Boswell. Rule induction with CN2: some recent improvements. In *EWSL*, pp. 151–163, 1991.

Martin C. Cooper and Leila Amgoud. Abductive explanations of classifiers under constraints: Complexity and properties. In *ECAI*, pp. 469–476, 2023.

William Craig. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *J. Symb. Log.*, 22(3):269–285, 1957.

Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In *ECAI*, pp. 712–720, 2020.

Adnan Darwiche and Chunxi Ji. On the computation of necessary and sufficient explanations. In *AAAI*, pp. 5582–5591, 2022.

Adnan Darwiche and Pierre Marquis. On quantifying literals in Boolean logic and its applications to explainable AI. *J. Artif. Intell. Res.*, 72:285–328, 2021.

Matteo Fischetti and Jason Jo. Deep neural networks and mixed integer linear optimization. *Constraints An Int. J.*, 23(3):296–309, 2018.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Niku Gorji and Sasha Rubin. Sufficient reasons for classifier decisions in the presence of domain constraints. In *AAAI*, pp. 5660–5667, 2022.

LLC Gurobi Optimization. Gurobi optimizer reference manual, 2026. URL http://www.gurobi.com.

Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva. On efficiently explaining graph-based classifiers. In *KR*, November 2021.

Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin C. Cooper, Nicholas Asher, and Joao Marques-Silva. Tractable explanations for d-DNNF classifiers. In *AAAI*, pp. 5719–5728, 2022.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NIPS*, pp. 4107–4115, 2016.

Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.

Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. Pysat: A python toolkit for prototyping with SAT oracles. In *SAT*, pp. 428–437, 2018.

Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. RC2: an efficient MaxSAT solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019a.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *AAAI*, pp. 1511–1519, 2019b.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. On relating explanations and adversarial examples. In *NeurIPS*, pp. 15857–15867, 2019c.

Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and Joao Marques-Silva. From contrastive to abductive explanations and back again. In *AI*IA*, pp. 335–355, 2020.

Alexey Ignatiev, Yacine Izza, Peter J. Stuckey, and Joao Marques-Silva. Using MaxSAT for efficient explanations of tree ensembles. In *AAAI*, pp. 3776–3785, 2022.

Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva. On explaining decision trees. *CoRR*, abs/2010.11034, 2020. URL https://arxiv.org/abs/2010.11034.

Xiayan Ji, Anton Xue, Eric Wong, Oleg Sokolsky, and Insup Lee. AR-Pro: Counterfactual explanations for anomaly repair with formal properties. In *NeurIPS*, 2024.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Comput. Surv.*, 55(5):95:1–95:29, 2023.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI machine learning repository. https://archive.ics.uci.edu, 2023. Accessed: 2026-01-14.

Faezeh Labbaf, Tomás Kolárik, Martin Blicha, Grigory Fedyukovich, Michael Wand, and Natasha Sharygina. Space explanations of neural network classification. In *CAV (3)*, pp. 287–303, 2025.

Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *KDD*, pp. 1675–1684. ACM, 2016.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *CoRR*, abs/1712.08443, 2017.

David Lewis. *Counterfactuals*. 1973.

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

Xinghan Liu and Emiliano Lorini. A unified logical framework for explanations in classifier systems. *J. Log. Comput.*, 33(2):485–515, 2023.

Emanuele La Malfa, Rhiannon Michelmore, Agnieszka M. Zbrzezny, Nicola Paoletti, and Marta Kwiatkowska. On guaranteed optimal robust explanations for NLP models. In *IJCAI*, pp. 2658–2665, 2021.

Joao Marques-Silva and Alexey Ignatiev. Delivering trustworthy AI through formal XAI. In *AAAI*, pp. 12342–12350, 2022.

Joao Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive Bayes and other linear classifiers with polynomial time and delay. In *NeurIPS*, 2020.

Joao Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska. Explanations for monotonic classifiers. In *ICML*, pp. 7469–7479, 2021.

Edward J McCluskey. Minimization of Boolean functions. *Bell system technical Journal*, 35(6): 1417–1444, 1956.

Kenneth L. McMillan. Interpolation and SAT-based model checking. In *CAV*, pp. 1–13, 2003.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2020. http://tiny.cc/6c76tz.

Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT**, pp. 607–617, 2020.

Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pp. 807–814, 2010.

Randal S. Olson, William G. La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Min.*, 10(1):36:1–36:13, 2017.

Willard V Quine. The problem of simplifying truth functions. *American mathematical monthly*, pp. 521–531, 1952.

Willard V Quine. A way to simplify truth functions. *American mathematical monthly*, pp. 627–631, 1955.

Ronald L. Rivest. Learning decision lists. *Mach. Learn.*, 2(3):229–246, 1987.

Joseph D. Romano, Trang T. Le, William G. La Cava, John T. Gregg, Daniel J. Goldberg, Natasha L. Ray, Praneel Chakraborty, Daniel S. Himmelstein, Weixuan Fu, and Jason H. Moore. PMLB v1.0: an open source dataset collection for benchmarking machine learning methods. *CoRR*, abs/2012.00058v2, 2021.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining Bayesian network classifiers. In *IJCAI*, pp. 5103–5111, 2018.

Ronal Singh, Tim Miller, Liz Sonenberg, Eduardo Velloso, Frank Vetere, Piers Howe, and Paul Dourish. An actionability assessment tool for explainable AI. *CoRR*, abs/2407.09516, 2024.

Robert C. Stalnaker. A theory of conditionals. In *Studies in Logical Theory*, pp. 98–112. 1968.

Christopher Umans. The minimum equivalent DNF problem and shortest implicants. In *FOCS*, pp. 556–563, 1998.

Christopher Umans, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Complexity of two-level logic minimization. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 25(7):1230–1246, 2006.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31 (2):841–887, 2018.

Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, Nina Narodytska, and Joao Marques-Silva. Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations. In *AAAI*, pp. 4123–4131, 2023.

A Experimental Details and Complete Results

This appendix gives the details of the experimental results omitted from Section 6. The supplementary material additionally contains the datasets, trained models, logs, raw CSV files, and scripts used to produce the results.

A.1 Benchmarks and Models

All datasets are public tabular classification benchmarks obtained from UCI or PMLB (Kelly et al., 2023; Olson et al., 2017; Romano et al., 2021). Table 3 provides the details of the datasets and the XGBoost and ReLU models trained in both experiments. Every original input column is treated as numerical. The same fixed 80%/20% split has been applied when training both models.

For the XGBoost experiment, every model uses 50 trees per class, an 80/20 train/test split, and the maximum depth shown in Table 3. The split uses the default seed value 7. The reported propositional encodings have 4–10183 variables and 304–102611 clauses. For the neural experiment, every feature is scaled to $[0, 1]$. A fully connected `sklearn` `MLPClassifier` with ReLU hidden layers $(16, 8)$ is trained on the training instances only using Adam, learning rate 0.005, maximum 1000 iterations, and seed 1234. Note that both XGBoost and ReLU models are trained on exactly the same training/test data, to illustrate the performance of CFXp on both.

Table 3: Benchmark and trained-model details. n and f are the numbers of rows and numerical features. d, tr., and te. denote XGBoost's maximum tree depth and train/test accuracy; k and p are the ReLU model's numbers of classes and trainable parameters.

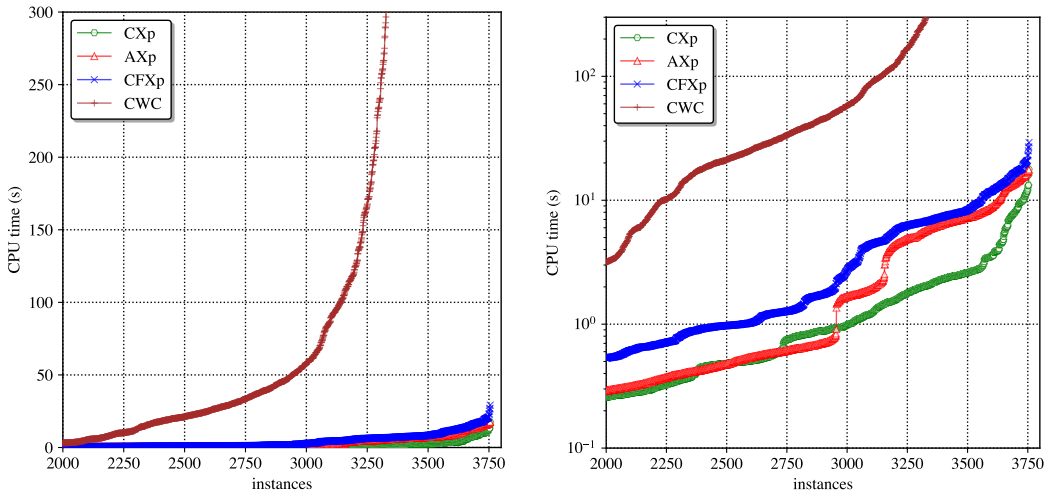
Dataset	n	f	d	XGB tr.	XGB te.	k	p	ReLU tr.	ReLU te.
ann-thyroid	7200	21	3	1.000	1.000	3	515	0.989	0.981
appendicitis	106	7	3	0.990	0.910	2	273	0.905	0.864
biodegradation	1055	41	3	0.970	0.870	2	817	0.922	0.872
divorce	170	54	3	1.000	1.000	2	1025	1.000	1.000
ecoli	327	7	3	1.000	0.850	5	309	0.904	0.909
glass2	163	9	4	1.000	0.880	2	305	0.892	0.848
ionosphere	351	34	3	1.000	0.930	2	705	0.989	0.944
pendigits	10992	16	3	1.000	0.990	10	498	0.992	0.984
promoters	106	58	3	1.000	1.000	2	1089	1.000	0.909
segmentation	210	19	3	1.000	0.950	7	519	0.994	0.857
shuttle	58000	9	3	1.000	1.000	7	359	0.998	0.997
sonar	208	60	3	1.000	0.860	2	1121	1.000	0.833
spambase	4601	57	4	0.980	0.960	2	1073	0.990	0.939
texture	5500	40	3	1.000	0.980	11	891	0.992	0.985
threeOf9	512	9	3	1.000	1.000	2	305	1.000	1.000
twonorm	7400	20	3	1.000	0.970	2	481	0.979	0.977
vowel	990	13	4	1.000	0.930	11	459	0.973	0.894
wdbc	569	30	4	1.000	0.970	2	641	0.982	0.991
wine-recognition	178	13	3	1.000	0.970	3	387	1.000	0.972
wpbc	194	33	4	1.000	0.740	2	689	1.000	0.692
zoo	101	16	4	1.000	0.830	7	471	1.000	1.000

A.2 Tree Ensembles: CWC Reference Experiment

The instances are sampled once by the original experiment with `random.seed(1234)`: up to 200 instances per dataset (or every instance for smaller datasets). This gives 3755 queries. AXp, CXp, and CFXp extraction uses the same MaxSAT encoding and incremental RC2 solver Ignatiev et al. (2018; 2019a) as the earlier MaxSAT study Ignatiev et al. (2022). A CFXp run first extracts a CXp and witness, then extracts an AXp for this witness, as described in Section 5. The five-minute limit applies independently to each formal explanation run; none of the AXp, CXp, or CFXp runs

Table 4: Aggregate formal explanation extraction results, for both XGBoost and ReLU models. Size and time report mean/median. CWC was run only on XGBoost models as it cannot be used for infinite feature domains. CWC size is the number of enumerated worlds on its 3327 completed runs; its time includes 300-second caps for 428 unfinished runs.

Model	Explanation	Size	Time (s)
XGBoost	AXp	9.043/ 8	1.529/0.266
XGBoost	CXp ($\|\gamma\|$)	1.972/ 1	0.783/0.184
XGBoost	CFXp support ($\|\delta\|$)	8.240/ 7	–
XGBoost	CFXp certificate ($\|\gamma \cup \delta\|$)	10.212/ 8	2.098/0.500
XGBoost	CWC worlds	117.770/19	51.872/2.168
ReLU	AXp	19.734/15	0.238/0.075
ReLU	CXp ($\|\gamma\|$)	2.064/ 1	0.064/0.033
ReLU	CFXp support ($\|\delta\|$)	20.577/17	–
ReLU	CFXp certificate ($\|\gamma \cup \delta\|$)	22.641/19	0.310/0.076



(a) Linear scale.

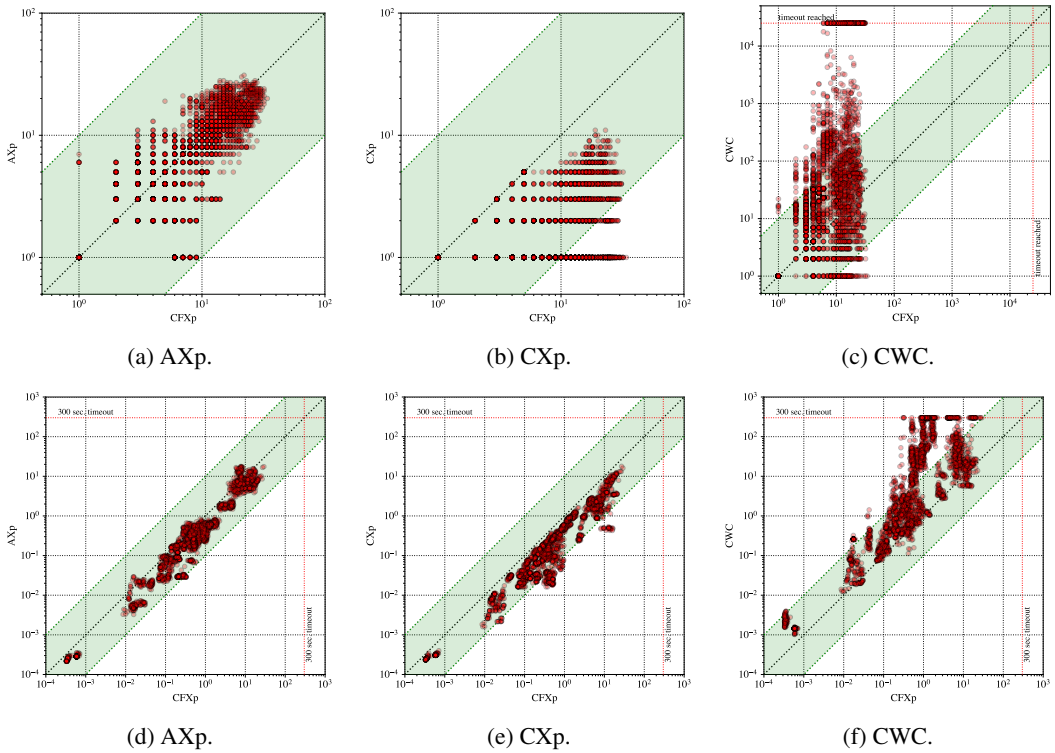
(b) Log scale.

Figure 1: Cactus plots for tree-ensemble extraction. CWC includes capped 300-second timeout observations.

timed out. Reported experiment times are process CPU times and exclude model encoding that is done once per dataset.

CWC builds on exhaustive enumeration of closest worlds rather than a single-witness algorithm. For each query, it enumerates all minimum-cardinality CXps and all their admissible witnesses over the finite partition induced by tree thresholds. Of the 3755 CWC computations, 3327 completed and 428 reached the 300-second limit. Its completed runs enumerate 1–17697 worlds (mean/median 117.770/19). The CWC mean in the main paper, 51.872 seconds, uses 300 seconds for every unfinished run. It is consequently a lower bound on the uncapped mean.

Figure 1 makes the cost of CWC visible. Although many CWC queries finish quickly, a larger number reaches the time limit, making CWC fall behind all the other formal approaches. The paired plots in Figure 2 give the full explanation size and runtime details behind the aggregate values shown in Section 6. They should not be read as a comparison of interchangeable output objects: CWC represents all closest worlds, whereas CFXp returns one witness together with a regional certificate.



(a) AXp. (b) CXp. (c) CWC.

(d) AXp. (e) CXp. (f) CWC.

Figure 2: Per-instance XGBoost comparisons. Top: explanation sizes, with CFXp measured by the joint certificate $|\gamma \cup \delta|$; bottom: CPU times.

A.3 Point-Explainer Setup for Tree Ensembles

The point methods use precisely the same trained XGBoost models and 3755 queries. DiCE 0.12 uses its random method with 1000 samples, post-hoc linear sparsity enhancement 0.1, and deterministic seed $1234 + 100 \cdot instance_id + target_class$. For each multiclass query, DiCE is run for every alternative class such that the model-verified point with lexicographically minimum L_0, normalized L_1, and normalized L_2 distance is picked. Its time includes all target searches.

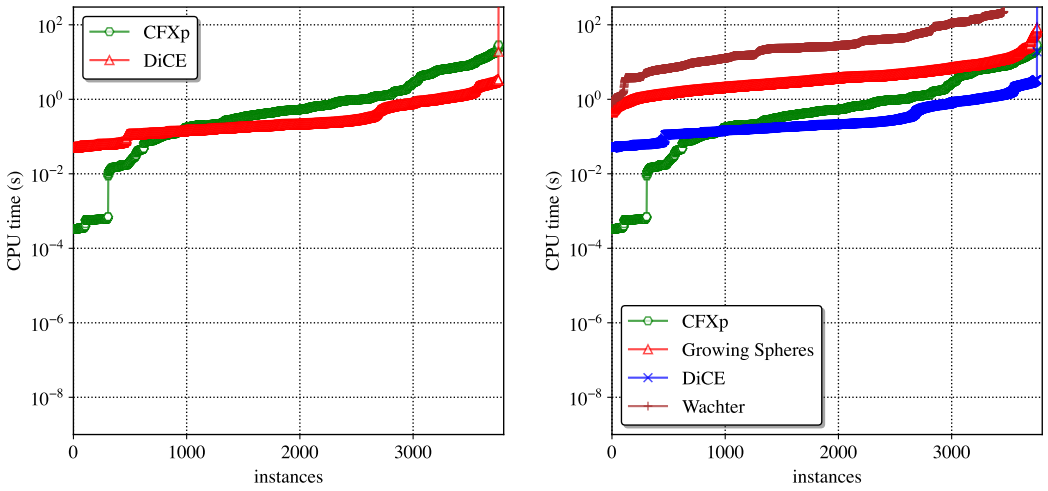
Growing Spheres uses the original authors' implementation at commit `894ce96`, with its documented defaults: 2000 samples per layer, ring layers, radius 0.1, decrease factor 10, untargeted search, and sparse projection.

Wachter uses `mlxtend.evaluate.create_counterfactual` from `mlxtend` 0.24.0. It scales each feature to $[0, 1]$, tries every alternative target and $\lambda \in \{0.1, 1, 10, 100, 1000\}$, and asks for target probability 1. The implementation's random initial point is restricted to rows already predicted as the intended target. Note that Wachter optimizes normalized L_1, not feature sparsity.

Every input column is numerical and all three point methods are run on all 3755 queries. Every returned point is re-evaluated by XGBoost and must be finite and within the observed feature ranges. Only these model-verified points contribute to the paired size statistics shown in the paper. CPU time is plotted in Figure 3. Initialization is excluded from both baseline and MaxSAT runtimes. Explanation size comparison is detailed in the scatter plots of Figure 4.

A.4 ReLU Networks Experiment

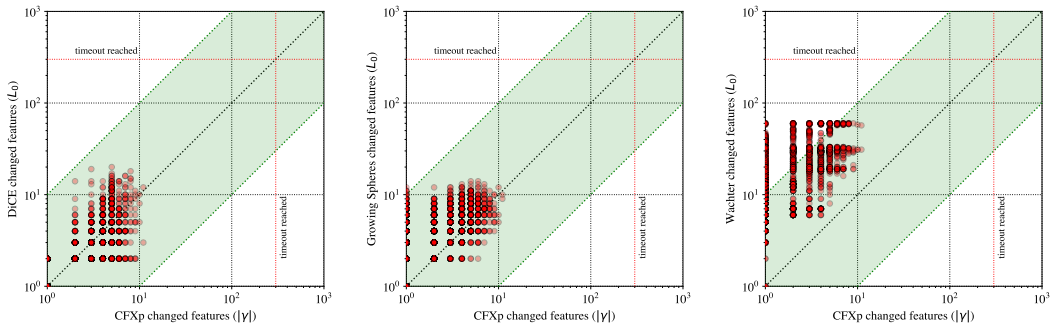
The neural experiment explains the decisions made by the trained ReLU models shown in Table 3 on the same raw query instances. Each original feature corresponds to one numerical MILP input with bounds $[0, 1]$. The backend follows the Fischetti–Jo ReLU formulation in mixed-integer linear programming (MILP) Fischetti & Jo (2018). For a ReLU, the encoding introduces a binary activation indicator and positive/negative parts. Class change is tested against each alternative output score



(a) All datasets: CFXp and DiCE.

(b) All datasets: all point methods.

Figure 3: XGBoost point-comparison runtime cactus plots. A point method that does not return a model-verified point is placed at the 300-second plot boundary.



(a) DiCE.

(b) Growing Spheres.

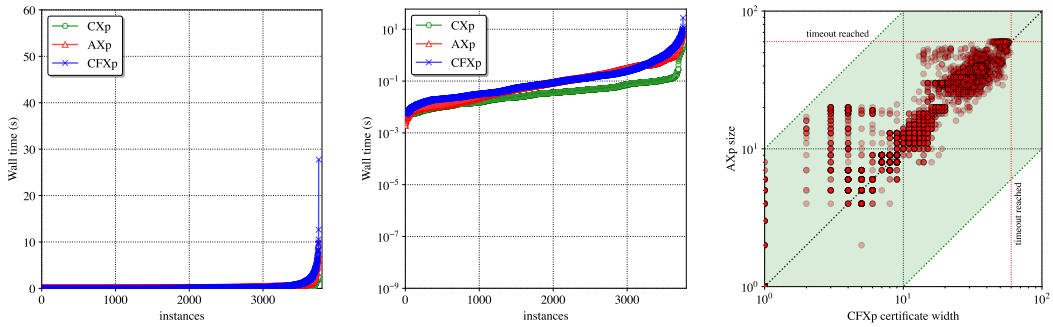
(c) Wachter.

Figure 4: XGBoost paired changed-feature statistics. Only model-verified point outputs are plotted; the horizontal axis is CFXp changed-feature set $|\gamma|$.

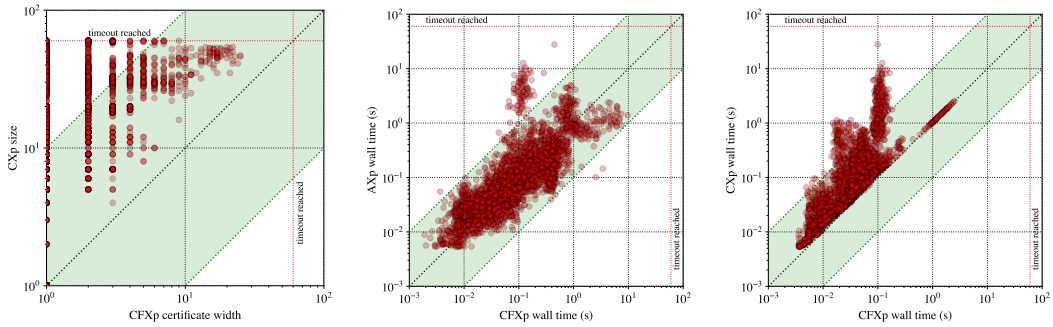
with margin 10^{-4}. Gurobi Gurobi Optimization (2026) uses one thread, a 60-second per-oracle-call limit, feasibility and integer-feasibility tolerances 10^{-8}, and disabled dual reductions. All 3755 formal explanations are successfully extracted, and every witness returned along the way for the SAT oracle calls is re-evaluated with the sklearn model.

Neural CXp, AXp, and CFXp extraction takes mean/median 0.064/0.033, 0.238/0.075, and 0.310/0.076 seconds, respectively. The corresponding mean/median oracle-call counts are 27.979/21 for CXps and 24.916/18 for AXp supports. The CFXp change set γ has range 1–25 and mean/median 2.064/1; the support δ and complete certificate $\gamma \cup \delta$ have ranges 0–59 and 1–60 and mean/median sizes 20.577/17 and 22.641/19. The larger supports are consistent with ordinary neural AXps, whose size is 19.734/15 on average/median.

As for the baselines, DiCE uses the same configuration as above and selects the best verified alternative target. Growing Spheres again uses the pinned authors' implementation and Wachter again uses `mlxtend` 0.24.0 with the same target and λ sweep. All three methods are run on the same 3755 queries. Candidate points are checked against the sklearn classifier, feature ranges, and the requested target. Unlike the XGBoost experiment, the times in Table 2 are wall-clock times for both neural CFXp extraction and point baselines. The full per-instance raw results and per-dataset summaries are provided.

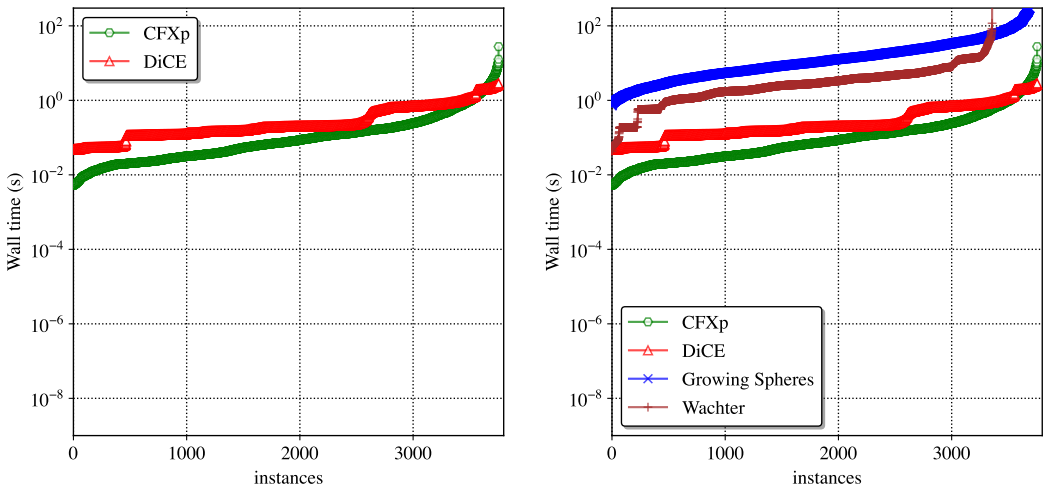


(a) Linear cactus plot. (b) Log cactus plot. (c) CFXp $|\gamma \cup \delta|$ vs. AXp size.



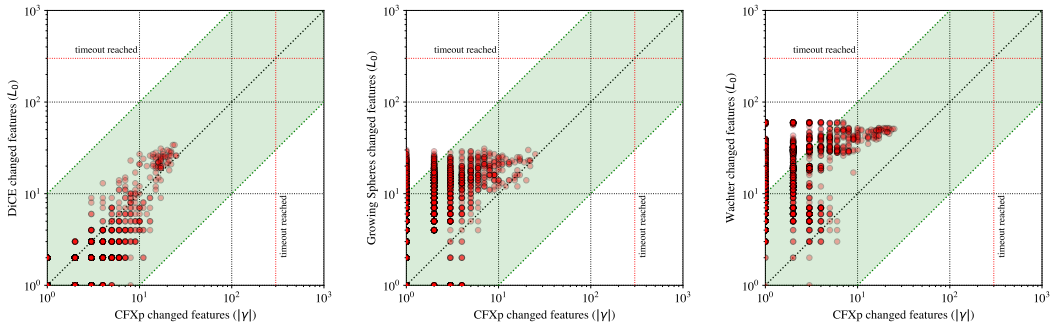
(d) CFXp $|\gamma \cup \delta|$ vs. CXp size. (e) CFXp versus AXp time. (f) CFXp versus CXp time.

Figure 5: Complete ReLU formal-extraction results, in the same style as the tree-ensemble plots. Times are single-threaded wall-clock times.



(a) All datasets: CFXp and DiCE. (b) All datasets: all point methods.

Figure 6: Runtime cactus plots for ReLU benchmarks. Calls producing points that are not verified by the model are placed at the 300-second plot boundary.



(a) DiCE. (b) Growing Spheres. (c) Wachter.

Figure 7: Comparison of explanation size for ReLU benchmarks. Only model-verified point outputs are plotted; the horizontal axis is $|\gamma|$.