Formal Explanations for Linear Temporal Logic over Finite Traces

Valeria Fionda, Antonio Ielo, Francesco Ricca

University of Calabria, Italy
valeria.fionda@unical.it, antonio.ielo@unical.it, francesco.ricca@unical.it

arXiv:2610.35169v1 [cs.LO] 26 Oct 2026

Abstract

Linear Temporal Logic over Finite Traces (LTL_f) is widely used in Artificial Intelligence to specify desired temporal behavior and determine whether finite traces satisfy or violate it. Explaining these outcomes is essential for debugging and trust. However, existing approaches typically capture either sufficient reasons or counterfactual causes, without clarifying their relationship. We introduce a native characterization of abductive (AXps) and contrastive (CXps) explanations directly over finite traces, based on *locking* and *unlocking* time-indexed events. We prove its equivalence to the classifier-based formulation, and show that CXps coincide with a recently proposed temporal causality semantics for LTL_f. This characterization enables an efficient implementation based on the enumeration of minimal unsatisfiable cores and minimal correction sets of ASP programs. Experiments on real-world event logs demonstrate the effectiveness of the approach.

Introduction

Linear Temporal Logic over Finite Traces (De Giacomo and Vardi 2013) (LTL_f) is a popular formalism for temporal reasoning in AI. It has been successfully applied to several domains, such as reinforcement learning (De Giacomo et al. 2019), planning (Bacchus and Kabanza 1998; De Giacomo and Rubin 2018), robotics (Kress-Gazit, Fainekos, and Pappas 2009) and business process management (van der Aalst, Pesic, and Schonenberg 2009; De Giacomo et al. 2014).

In applications, LTL_f formulas are used to describe *desired behavior*. A formula may describe the dynamics of a system, or the goals driving an agent's behavior. Traces (interpretations) represents states of a system over time, or a sequence of actions performed by an agent. When a trace π satisfies a formula φ, we say that π *complies* with the specification; otherwise, it *violates* it. In either case, being able to explain *why* is important.

Figure 1 illustrates the explanation problem via an autonomous delivery agent subject to $\varphi = \mathsf{G}\ (req \rightarrow \mathsf{F}\ deliver) \wedge \mathsf{G}\ (low \rightarrow \mathsf{X}\ charge)$, which requires every request to be eventually served and every low-battery to be immediately followed by charging. The trace π_1 (of Figure 1) satisfies φ, but this alone does not explain *why*. We focus on two complementary kinds of explanation. A *sufficient reason* is a set of time-indexed events and non-events that guarantees satisfaction: for π_1, the absence of low throughout the trace already guarantees satisfaction of the charging obligation, no matter how the remaining observations are altered. A *minimal intervention* is a minimal set of observations whose change flips the outcome: adding low to the first time instant of π not followed by $charge$ turns it into a violating trace. Conversely, π_2 violates φ: here a sufficient reason guarantees the violation (e.g., knowing only that low holds at the second time instant and $charge$ is absent at the third time instant makes *every* trace agreeing with these two observations violate φ), while minimal interventions restore satisfaction. Thus the explanation problem answers two complementary questions: *why does the trace satisfy or violate the specification?* and *what must change to obtain the opposite outcome?*

These two kinds of explanation are precisely the central notions of formal explainable AI (FXAI) (Ignatiev, Narodytska, and Marques-Silva 2019; Shih, Choi, and Darwiche 2018): *abductive* (AXps) and *contrastive* (CXps) explanations. Given a classifier and an input instance, an abductive explanation is a subset of feature values sufficient to guarantee the predicted class, whereas a contrastive explanation is a subset of features whose change yields a different class. Importantly, subset-minimal abductive and contrastive explanations are related by hitting-set duality (Ignatiev et al. 2020; Reiter 1987), which also supports effective enumeration algorithms (Liffiton et al. 2016). In principle, an LTL_f formula can be cast as a binary classifier over time-indexed propositions, making these notions formally applicable. However, this encoding is not the usual way temporal formulas are understood and makes it harder to see explanations as observations that are kept fixed or allowed to change. Thus, a *native characterization* of AXps and CXps for LTL_f is missing. Existing approaches capture either sufficient reasons for satisfaction (Ferrère, Maler, and Ničković 2015) or counterfactual causes of violation (Meng and Alrajeh 2025), but none connects the two perspectives or grounds them in hitting-set duality. We fill this gap through the simple operations of *locking* and *unlocking* temporal observations: locked observations play the role of the feature values an AXp keeps fixed, whereas unlocked ones are those a CXp is allowed to alter. AXps and CXps are thus defined directly over finite traces, with no classifier-based reformulation. Our contributions are threefold: *(i)* we characterize AXps and CXps for the satisfaction and violation of LTL_f specifications, directly over finite traces; *(ii)* we prove that this characterization is

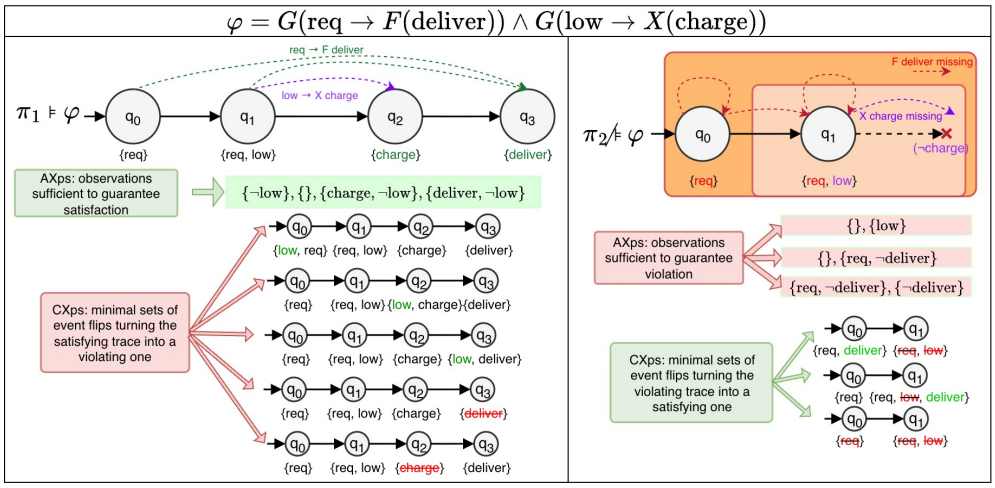


Figure 1: AXps (sufficient reasons) and CXps (minimal outcome-flipping interventions) for φ on a compliant trace π_1 and a violating trace π_2. Sets are listed state by state; e.g. $\{\}, \{low\}$ denotes $\{low_1\}$.

equivalent to the classifier-based one, establish the hitting-set duality between AXps and CXps for LTL_f, and show that CXps coincide with the temporal causes of (Meng and Alrajeh 2025) under fixed-length interventions; *(iii)* we reduce their computation to the enumeration of minimal unsatisfiable subsets and minimal correction sets, implement the approach in Answer Set Programming (ASP) (Gelfond and Lifschitz 1991; Brewka, Eiter, and Truszczynski 2011), and evaluate it on real-world event logs.

Related Work

Formal Explanations. Our work instantiates the general framework of abductive and contrastive explanations (Shih, Choi, and Darwiche 2018; Ignatiev, Narodytska, and Marques-Silva 2019; Ignatiev et al. 2020; Darwiche and Hirth 2023) in the setting of LTL_f. However, the standard formulation treats the object being explained as a classifier over features. While this viewpoint is formally applicable to LTL_f, it is not natural for temporal-logic practitioners. We therefore provide an equivalent characterization stated directly in terms of LTL_f formulas and finite traces. Similar specializations have been developed for settings in which the object being explained is not naturally represented as a classifier, such as ranking functions (Chiariello and Marques-Silva 2025) and finite-state machines (Granada, Ignatiev, and Stuckey 2026). Our work follows the same direction by providing a specialization for LTL_f formulas.

Temporal Implicants & Temporal Causality. Implicants are partial interpretations guaranteeing a propositional formula's truth or falsity and, for propositional binary classifiers, coincide with abductive explanations (Ignatiev et al. 2020). In LTL, temporal implicants have only been studied under infinite-trace semantics (Ferrère, Maler, and Ničković 2015), and implemented with a greedy algorithm computing a single, possibly non-minimal implicant. The connection to unsatisfiable cores is mentioned as a related problem but not further developed. We instead address finite traces, recast the problem in the FXAI framework, and provide an efficient implementation based on unsatisfiable cores and correction sets.

A recent work by (Meng and Alrajeh 2025) proposes a revised version of *cause* for LTL_f, specializing the Halpern's (Halpern 2016). It extends earlier works (Beer et al. 2009) that considers only singleton causes, to conjunctive causes. Neither line addresses the dual notion of sufficient reasons for satisfaction or violation. We will prove that this notion is closely related to CXps.

Unsatisfiable Cores in Temporal Reasoning. Minimal unsatisfiable cores and correction sets (Liffiton and Sakallah 2008; Marques-Silva, Janota, and Mencía 2017) are at the base of our implementation. LTL_f admits notions of unsatisfiable cores (Roveri et al. 2024), whose computation has been the focus of recent works (Roveri et al. 2024; Niu et al. 2024; Fionda, Ielo, and Ricca 2026; Ielo et al. 2026). Unsatisfiable cores explain *why a formula is unsatisfiable*, whereas abductive explanations answer *why a trace violates/satisfies a formula*. The first is a form of *global* explanation, the latter is a form of *local* explanation (Guidotti et al. 2019; Marques-Silva and Ignatiev 2022). Our implementation is based on bounded LTL_f satisfiability (Fionda, Ielo, and Ricca 2024), obtaining AXps and CXps as unsatisfiable cores and correction sets of a logic program (Alviano et al. 2023). The underlying solver is based on SAT techniques (Liffiton et al. 2016).

Answer Set Programming

An *atom* is an expression of the form $p(t_1, \ldots, t_n)$ where p is a predicate, $t_1, \ldots, t_n$ are terms (i.e., constants, integer numbers, strings or variables), and n is known as *arity* of the predicate. A *literal* is either an atom a or its *negation* $not\ a$, where not denotes the negation as failure. A *rule* r is an expression of the form $h \leftarrow b_1, \ldots, b_n$ where h is an atom

referred to as *head*, denoted by H_r, that can also be omitted, $n \geq 0$, and $b_1, \ldots, b_n$ is a conjunction of literals referred to as *body*, denoted by B_r. In particular, a rule is said to be a *constraint* if its head is omitted, while it is said to be a *fact* if $n = 0$. A rule r is *safe* if each variable appears at least in one positive literal in the body of r. A *program* in ASP is a finite set of safe rules. Given a program P, and $r \in P$, $ground(r)$ is the set of ground instantiations of r obtained by replacing variables in r with constants in P; whereas $ground(P)$ is the union of ground instantiations of rules in P. Given a program P, the Herbrand's base of P, denoted by $\mathcal{B}_P$, is the set of atoms constructible from constants and predicate names occurring in P. An *interpretation* I for P is a subset of $\mathcal{B}_P$, which is an answer set of P if (i) I is a model, i.e., for each rule $r \in ground(P)$ either the head of r is true wrt I or the body of r is false wrt I; and (ii) I is a minimal model of its GL-reduct (Gelfond and Lifschitz 1991).

Minimal Unsatisfiable Subprograms. Consider a program P and a set of objective atoms $O \subseteq \mathcal{B}_P$. For $S \subseteq O$, enforce(P, O, S) is the program obtained from P by adding a choice over atoms in O (i.e., o_i :- $not\ o_i$ no_i :- o_i, for $i \in [1, n]$) and a set of constraints of the form :- $not\ o$, for every $o \in S$. Intuitively, enforce(P, O, S) augments the program P in such a way that the objective atoms can be arbitrarily chosen (i.e. either as true or false) but the atoms in S are *enforced* to be true. An *unsatisfiable subset* for P wrt the set of objective atoms O is a set of atoms $U \subseteq O$ such that enforce(P, O, U) is incoherent (Alviano et al. 2023). $US(P, O)$ denotes the set of unsatisfiable subsets of P wrt O. An unsatisfiable subset $U \in US(P, O)$ is a *minimal unsatisfiable subset* (MUS) of P wrt O iff for every $U' \subset U$, $U' \notin US(P, O)$. An answer set M of P is minimal wrt $O \subset \mathcal{B}_\mathcal{P}$ iff there exist no other answer set M' of P such that $(M \cap O) \subset (M' \cap O)$ (Alviano et al. 2023).

More details on ASP are available in (Gelfond and Lifschitz 1991; Brewka, Eiter, and Truszczynski 2011; Gebser et al. 2012; Lifschitz 2019).

Linear Temporal Logic on Finite Traces

Linear Temporal Logic on Finite Traces (LTL_f) (De Giacomo and Vardi 2013) is a variant of Linear Temporal Logic (LTL) (Pnueli 1977) where the models, called *traces*, are assumed to be finite. An LTL_f formula over the alphabet $\mathcal{A}$ is defined according to the following grammar:

$$\varphi ::= a \in \mathcal{A} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

where X ("Next") and U ("Until") are temporal operators. Standard propositional boolean ($\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$), and temporal (e.g., *Eventually* F, *Always* G, *Weak Next* X_w and *Release* R) shorthands are defined as usual. An interpretation is a finite trace $\pi = \pi_0\pi_1 \cdots \pi_{n-1}$, with $\pi_i \subseteq \mathcal{A}$. $|\pi| = n$ is the trace length, and $\pi(i) = \pi_i$ is the i-th state of π. The satisfaction relation $\pi, i \models \varphi$ is defined inductively as follows:

$\pi, i \models a$ iff $a \in \pi(i)$
$\pi, i \models \neg\varphi$ iff $\pi, i \not\models \varphi$
$\pi, i \models \varphi \wedge \psi$ iff $\pi, i \models \varphi$ and $\pi, i \models \psi$
$\pi, i \models \mathsf{X}\,\varphi$ iff $i + 1 < |\pi|$ and $\pi, i + 1 \models \varphi$
$\pi, i \models \varphi\,\mathsf{U}\,\psi$ iff there exists k with $i \leq k < |\pi|$ such that $\pi, k \models \psi$ and for all j with $i \leq j < k$, $\pi, j \models \varphi$

π is a *model* of φ if $\pi, 0 \models \varphi$, denoted as $\pi \models \varphi$. We denote by $\mathsf{Mod}(\varphi)$ the set of models of φ, that is $\mathsf{Mod}(\varphi) = \{\pi \in (2^\mathcal{A})^* : \pi \models \varphi\}$. The satisfiability problem for LTL_f, i.e., deciding whether there exists a trace π such that $\pi \models \varphi$, is PSPACE-complete (De Giacomo and Vardi 2013). Complexity drops to NP-complete restricting the problem to traces of bounded length (Fionda and Greco 2018).

Formal Explanations

Let $\mathcal{F}$ be a finite set of *features*. Each feature $i \in \mathcal{F}$ is associated to a finite *domain* D_i. The corresponding *feature space* $\mathbb{F}$ is the Cartesian product of the feature domains, $\mathbb{F} = \prod_{i \in \mathcal{F}} D_i$. Each element $\alpha \in \mathbb{F}$ is a total assignment of values to the features, with $\alpha(i) \in D_i$ for every $i \in \mathcal{F}$. We refer to such an assignment as an *instance*. A classifier is a total function $f : \mathbb{F} \mapsto \mathcal{K}$, where $\mathcal{K}$ is a finite set of class labels. Given $\mathcal{S} \subseteq \mathcal{F}$ and two instances $\alpha, \beta \in \mathbb{F}$, we say α and β are *compatible* in $\mathcal{S}$ and write $\alpha \equiv_\mathcal{S} \beta$ if, and only if, $\alpha(i) = \beta(i)$ for every $i \in \mathcal{S}$.

An *explanation problem* is a tuple (f, α, c), where $f : \mathbb{F} \mapsto \mathcal{K}$ is a classifier, $\alpha \in \mathbb{F}$ is an instance, and $c \in \mathcal{K}$ is a class label such that $f(\alpha) = c$. Two main types of formal explanations are in the literature: *abductive* (AXp) and *contrastive* (CXp) explanations. An abductive explanation for (f, α, c) answers the question *Why was class c assigned to α?*, while a contrastive explanation addresses the question *Why was α not assigned a class different from c?*.

Weak Abductive Explanations (WAXp). We say that $\mathcal{X} \subseteq \mathcal{F}$ is a WAXp for (f, α, c) if the following holds:

$$\forall \beta \in \mathbb{F}.\ \beta \equiv_\mathcal{X} \alpha \implies f(\beta) = c$$

Weak Contrastive Explanations (WCXp). We say that $\mathcal{Y} \subseteq \mathcal{F}$ is a WCXp for (f, α, c) if the following holds:

$$\exists \beta \in \mathbb{F}.\ \beta \equiv_{\mathcal{F} \setminus \mathcal{Y}} \alpha \wedge f(\beta) \neq c$$

Intuitively, an abductive explanation identifies a set of feature values that is sufficient to guarantee the prediction c, that is once the feature in $\mathcal{X}$ are fixed to the values they assume in α, every compatible instance is still classified as c. Instead, a contrastive explanation identifies a set of features whose values can be changed to alter the prediction, that is, keeping all features outside $\mathcal{Y}$ fixed to the values in α, there exists an assignment to the features in $\mathcal{Y}$ for which the classifier outputs a class different from c. Subset-minimal WAXp and WCXp are called abductive explanations (AXp) and contrastive explanations (CXp), respectively. AXps and CXps are related by *hitting-set duality*: every AXp is a minimal hitting set of the set of all CXps, and vice versa (Ignatiev et al. 2020).

AXp and CXp for LTL_f

In this section, we reformulate abductive and contrastive explanations for LTL_f. In the rest of the paper, we fix a finite

alphabet $\mathcal{A}$ and a trace length $n \in \mathbb{N}$. We assume all traces are sequences in $(2^{\mathcal{A}})^n$.

Notation. We adopt a set-based representation of traces based on *temporal literals*. Let $\mathsf{Vars} = \{a_i : a \in \mathcal{A},\ 0 \leq i < n\}$ denote the set of time-indexed atoms, where a_i indicates that a holds at time instant i. The set of temporal literals is defined as $\mathsf{Lits} = \bigcup_{a_i \in \mathsf{Vars}} \{a_i, \neg a_i\}$. For every temporal literal $\ell \in \mathsf{Lits}$, we denote by $\mathsf{var}(\ell)$ its underlying time-indexed atom, i.e., $\mathsf{var}(a_i) = a_i$ and $\mathsf{var}(\neg a_i) = a_i$. Similarly, $\overline{\ell}$ denotes the opposite literal, that is, $\overline{a_i} = \neg a_i$ and $\overline{\neg a_i} = a_i$. A set of temporal literals $E \subseteq \mathsf{Lits}$ is said to be *invalid* if it contains both a literal and its complement, *valid* otherwise, and *complete* if, for every $a_i \in \mathsf{Vars}$, exactly one of a_i or $\neg a_i$ belongs to E. A trace π is represented by the valid, complete set of temporal literals:

$$\mathsf{L}(\pi) = \bigcup_i \left(\{a_i : a \in \pi(i)\} \cup \{\neg a_i : a \notin \pi(i)\}\right).$$

For instance, the violating trace π_2 of Figure 1, of length $n = 2$, is represented by $\mathsf{L}(\pi_2) = \{req_0, req_1, low_1\} \cup \{\neg deliver_0, \neg low_0, \neg charge_0, \neg deliver_1, \neg charge_1\}$.
Given a set of temporal literals E, two traces π and τ are *compatible* on E if they agree on every literal in E, that is, $E \subseteq \mathsf{L}(\pi) \cap \mathsf{L}(\tau)$. Accordingly, we define the set of all traces compatible with π on E as:

$$\mathsf{Comp}_\pi(E) = \{\tau \in (2^{\mathcal{A}})^{|\pi|} : E \subseteq \mathsf{L}(\pi) \cap \mathsf{L}(\tau)\}.$$

Intuitively, $\mathsf{Comp}_\pi(E)$ consists of all traces having the same length of π that preserve the behavior of π specified by E in terms of events happening (positive literals) and not happening (negative literals). Since E is not required to be complete, only the events explicitly described by its literals are fixed, while all remaining propositions at all time instants are left unconstrained.

Explanation problem. An *explanation problem* is a tuple (π, φ), where $\pi \in (2^{\mathcal{A}})^n$ is a trace and φ is an LTL_f formula. We now reformulate weak abductive and weak contrastive explanations using the notation introduced above. Since $\pi \not\models \varphi \Leftrightarrow \pi \models \neg\varphi$, we focus on explanation settings (π, φ) such that $\pi \not\models \varphi$ without loss of generality. The complementary case is handled by replacing φ with $\neg\varphi$. Intuitively, a set $E \subseteq \mathsf{L}(\pi)$ identifies the temporal literals that are *locked*, i.e., forced to retain the truth value they have in π, while all literals outside E are *unlocked* and free to change: $\mathsf{Comp}_\pi(E)$ is precisely the set of traces obtained by locking E. An abductive explanation locks enough observations to make the violation unavoidable; a contrastive explanation unlocks enough observations to make satisfaction possible.

Definition 1 (WAXp)**.** Let (π, φ) be an explanation problem. We say that $E \subseteq \mathsf{L}(\pi)$ is a WAXp for (π, φ) if: $\pi \not\models \varphi$ and $\mathsf{Comp}_\pi(E) \cap \mathsf{Mod}(\varphi) = \emptyset$ ■

Definition 2 (WCXp)**.** Let (π, φ) be an explanation problem. We say that $E \subseteq \mathsf{L}(\pi)$ is a WCXp for (π, φ) if: $\pi \not\models \varphi$ and $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus E) \cap \mathsf{Mod}(\varphi) \neq \emptyset$ ■

Definition 3 (AXp, CXp)**.** Let (π, φ) be an explanation problem. A set $E \subseteq \mathsf{L}(\pi)$ is an *AXp* (resp. *CXp*) for (π, φ) if it is a WAXp (resp. WCXp) and no $E' \subset E$ is a WAXp (resp. WCXp). ■

Example 1. Consider π_2 and φ from Figure 1. The set $E = \{low_1\}$ is an AXp for (π_2, φ): locking low_1 is enough, since every trace of length 2 in which low holds at the last instant violates $\mathsf{G}\,(low \rightarrow \mathsf{X}\,charge)$, regardless of all other observations; and E is minimal, as the empty set is not a WAXp. Dually, $\{\neg deliver_0, req_1, low_1\}$ is a CXp: unlocking these three literals admits a model of φ, where $deliver$ is added at instant 0 and req, low are removed at instant 1, and no proper subset is a WCXp. ◁

Equivalence to Classifier Explanations

In this section, we formalize the LTL_f-as-binary-classifier setting, establish a correspondence between classifier-based explanations and our reformulation, and prove that the two formulations are equivalent.

Encoding LTL_f Formulas as Binary Classifiers

Let $\mathcal{F} = \mathsf{Vars}$ be binary features, each with domain $\{\bot, \top\}$– that is, we associate one Boolean feature with each time-indexed atom a_i–and let $\mathcal{K} = \{0, 1\}$. A trace $\pi \in (2^{\mathcal{A}})^n$ is represented by the assignment α_π defined by $\alpha_\pi(a_i) = \top \Leftrightarrow a \in \pi(i)$. We denote by $\mathbb{F}$ the set of such assignments. Conversely, every assignment $\alpha \in \mathbb{F}$ introduces a trace π_α, that is $\pi_\alpha(i) = \{a \in \mathcal{A} : \alpha(a_i) = \top\}$.

Every LTL_f formula φ induces a binary classifier $f_\varphi : \mathbb{F} \mapsto \{0, 1\}$, defined by

$$f_\varphi(\alpha) = \begin{cases} 1 & \text{if } \pi_\alpha \models \varphi, \\ 0 & \text{otherwise.} \end{cases}$$

As in the previous section, we restrict our attention to instances satisfying $f_\varphi(\alpha_\pi) = 0$, since $\pi \not\models \varphi$ iff $\pi \models \neg\varphi$.

A set $\mathcal{X} \subseteq \mathcal{F}$ is a WAXp for $f_\varphi(\alpha_\pi) = 0$ if:

$$\forall(\alpha \in \mathbb{F}).\ \alpha \equiv_{\mathcal{X}} \alpha_\pi \implies f_\varphi(\alpha) = 0.$$

A set $\mathcal{Y} \subseteq \mathcal{F}$ is a WCXp for $f_\varphi(\alpha_\pi) = 0$ if:

$$\exists(\alpha \in \mathbb{F}).\ \alpha \equiv_{\mathcal{F} \setminus \mathcal{Y}} \alpha_\pi \ \wedge f_\varphi(\alpha) = 1.$$

Both formulations use the same names; they are distinguished by their arguments, as a WAXp (WCXp) for $f_\varphi(\alpha_\pi) = 0$ is a set of *features*, whereas a WAXp (WCXp) for (π, φ), as per Definitions 1 and 2, is a set of *temporal literals*. When needed, we call the former *classifier-based* and the latter *trace-based*.

To relate the two formulations, we introduce two mappings between sets of features and sets of temporal literals. Given a trace π and a set of feature $\mathcal{S} \subseteq \mathcal{F}$, we denote by $\mathsf{lit}_\pi(\mathcal{S})$ the set of literals of π over the features in $\mathcal{S}$, that is,

$$\mathsf{lit}_\pi(\mathcal{S}) = \{\ell \in \mathsf{L}(\pi) : \mathsf{var}(\ell) \in \mathcal{S}\}.$$

Conversely, for a set of temporal literals $E \subseteq \mathsf{L}(\pi)$, we denote by $\mathsf{feat}(E)$ the corresponding set of features:

$$\mathsf{feat}(E) = \{\mathsf{var}(\ell) : \ell \in E\}.$$

Observe that, by construction, for every trace π and every $\mathcal{S} \subseteq \mathcal{F}$, the following properties hold:

1. $\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{S})) = \mathcal{S}$;
2. $\mathsf{lit}_\pi(\mathsf{feat}(E)) = E$, and
3. $\mathsf{L}(\pi) \setminus \mathsf{lit}_\pi(\mathcal{S}) = \mathsf{lit}_\pi(\mathcal{F} \setminus \mathcal{S})$.

The equivalence proofs rely on the following observation: agreement between assignments corresponds to compatibility between the corresponding traces. More precisely, if two assignments agree on a set of features $\mathcal{S}$, then the corresponding traces share the temporal literals in $\mathsf{lit}_\pi(\mathcal{S})$.

Proposition 1. *Let π, τ be traces and let $E \subseteq \mathsf{L}(\pi)$. Then*

$$\tau \in \mathsf{Comp}_\pi(E) \quad \textit{if and only if} \quad \alpha_\tau \equiv_{\mathsf{feat}(E)} \alpha_\pi.$$

Proof. ($\rightarrow$) Assume that $\tau \in \mathsf{Comp}_\pi(E)$. By definition, $E \subseteq \mathsf{L}(\pi) \cap \mathsf{L}(\tau)$. Hence, every temporal literal in E belongs to both $\mathsf{L}(\pi)$ and $\mathsf{L}(\tau)$. Since $E \subseteq \mathsf{L}(\pi)$, every feature in $\mathsf{feat}(E)$ is assigned the same truth value by α_π and α_τ. Therefore, $\alpha_\tau \equiv_{\mathsf{feat}(E)} \alpha_\pi$. ($\leftarrow$) Assume that $\alpha_\tau \equiv_{\mathsf{feat}(E)} \alpha_\pi$. Then, for every feature $a_i \in \mathsf{feat}(E)$, the assignments α_π and α_τ are the same. Since $E \subseteq \mathsf{L}(\pi)$, each literal in E also belongs to $\mathsf{L}(\tau)$. Thus, $E \subseteq \mathsf{L}(\pi) \cap \mathsf{L}(\tau)$, which implies that $\tau \in \mathsf{Comp}_\pi(E)$. □

Using Proposition 1, we can now prove the equivalence of the two notions of explanation. We start with WAXps.

Proposition 2. *Let (π, φ) be an explanation problem such that $\pi \not\models \varphi$, and let $\mathcal{X} \subseteq \mathcal{F}$. Then $\mathcal{X}$ is a WAXp for $f_\varphi(\alpha_\pi) = 0$ if and only if $\mathsf{lit}_\pi(\mathcal{X})$ is a WAXp for (π, φ).*

Proof. ($\rightarrow$) Assume that $\mathcal{X}$ is a WAXp for $f_\varphi(\alpha_\pi) = 0$, and let $\tau \in \mathsf{Comp}_\pi(\mathsf{lit}_\pi(\mathcal{X}))$. By Proposition 1, $\alpha_\tau \equiv_{\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{X}))} \alpha_\pi$. Since $\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{X})) = \mathcal{X}$, we obtain $\alpha_\tau \equiv_{\mathcal{X}} \alpha_\pi$. Because $\mathcal{X}$ is a WAXp, $f_\varphi(\alpha_\tau) = 0$, which implies $\tau \not\models \varphi$. Therefore, $\mathsf{Comp}_\pi(\mathsf{lit}_\pi(\mathcal{X})) \cap \mathsf{Mod}(\varphi) = \emptyset$, and hence $\mathsf{lit}_\pi(\mathcal{X})$ is a WAXp for (π, φ). ($\leftarrow$) Assume that $\mathsf{lit}_\pi(\mathcal{X})$ is a WAXp for (π, φ). Let $\alpha \in \mathbb{F}$ be such that $\alpha \equiv_{\mathcal{X}} \alpha_\pi$. Let $\tau = \pi_\alpha$. Since $\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{X})) = \mathcal{X}$, Proposition 1 implies that $\tau \in \mathsf{Comp}_\pi(\mathsf{lit}_\pi(\mathcal{X}))$. Because $\mathsf{lit}_\pi(\mathcal{X})$ is a WAXp for (π, φ), $\tau \not\models \varphi$. Hence $f_\varphi(\alpha) = 0$, which proves that $\mathcal{X}$ is a WAXp for $f_\varphi(\alpha_\pi) = 0$. □

An analogous correspondence holds for WCXp.

Proposition 3. *Let (π, φ) be an explanation problem such that $\pi \not\models \varphi$ and let $\mathcal{Y} \subseteq \mathcal{F}$. Then $\mathcal{Y}$ is a WCXp for $f_\varphi(\alpha_\pi) = 0$ if and only if $\mathsf{lit}_\pi(\mathcal{Y})$ is a WCXp for (π, φ).*

Proof. ($\rightarrow$) Assume that $\mathcal{Y}$ is a WCXp for $f_\varphi(\alpha_\pi) = 0$. Then there exists $\alpha \in \mathbb{F}$ such that $\alpha \equiv_{\mathcal{F} \setminus \mathcal{Y}} \alpha_\pi$ and $f_\varphi(\alpha) = 1$. Let $\tau = \pi_\alpha$. By Proposition 1, $\tau \in \mathsf{Comp}_\pi(\mathsf{lit}_\pi(\mathcal{F} \setminus \mathcal{Y}))$ that is equivalent to $\tau \in \mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus \mathsf{lit}_\pi(\mathcal{Y}))$ (property 3 above). Moreover, $f_\varphi(\alpha) = 1$ implies $\tau \models \varphi$. Hence $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus \mathsf{lit}_\pi(\mathcal{Y})) \cap \mathsf{Mod}(\varphi) \neq \emptyset$, which proves that $\mathsf{lit}_\pi(\mathcal{Y})$ is a WCXp for (π, φ). ($\leftarrow$) Assume $\mathsf{lit}_\pi(\mathcal{Y})$ being a WCXp for (π, φ). Then there exists $\tau \in \mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus \mathsf{lit}_\pi(\mathcal{Y})) \cap \mathsf{Mod}(\varphi)$. By property 3 above, $\tau \in \mathsf{Comp}_\pi(\mathsf{lit}_\pi(\mathcal{F} \setminus \mathcal{Y}))$. By Proposition 1, $\alpha_\tau \equiv_{\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{F} \setminus \mathcal{Y}))} \alpha_\pi$. Since $\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{F} \setminus \mathcal{Y})) = \mathcal{F} \setminus \mathcal{Y}$ (property 1), we have $\alpha_\tau \equiv_{\mathcal{F} \setminus \mathcal{Y}} \alpha_\pi$. Finally, $\tau \models \varphi$ implies $f_\varphi(\alpha_\tau) = 1$, and thus $\mathcal{Y}$ is a WCXp for $f_\varphi(\alpha_\pi) = 0$. □

The previous propositions establish a bijective correspondence between the two formulations, i.e. each set $E \subseteq \mathsf{L}(\pi)$ that is a WAXp (WCXp, respectively) for (π, φ) can be decoded into the set $\mathsf{feat}(E) \subseteq \mathcal{F}$ that is a WAXp (WCXp, respectively) for $f_\varphi(\alpha_\pi) = 0$. The result holds since $\mathsf{feat}(\mathsf{lit}_\pi(\mathcal{S})) = \mathcal{S}$ and $\mathsf{lit}_\pi(\mathsf{feat}(E)) = E$. Moreover, from the fact that both preserve subsets, we have that the statements also hold for AXp and CXp.

Corollary 1. *Let (π, φ) be an explanation problem with $\pi \not\models \varphi$. Then $\mathcal{X}$ is an AXp (resp. CXp) for $f_\varphi(\alpha_\pi) = 0$ if and only if $\mathsf{lit}_\pi(\mathcal{X})$ is an AXp (resp. CXp) for (π, φ).*

By AXp/CXp duality (Ignatiev et al. 2020) and since $\mathsf{lit}_\pi(\cdot)$ is an intersection- and inclusion-preserving bijection between $\mathcal{F}$ and $\mathsf{L}(\pi)$, the following result holds.

Corollary 2 (Duality)**.** *Let (π, φ) be an explanation problem with $\pi \not\models \varphi$. Then every AXp for (π, φ) is a minimal hitting set of the set of all CXps for (π, φ), and viceversa.*

Relationship with Temporal Causality

Meng et al. (Meng and Alrajeh 2025) recently proposed a causality-based framework for explaining finite counterexamples to LTL_f specifications. Their notion of explanation is based on Halpern's causality, that is a cause is a minimal set of interventions that transform a counterexample trace into a model of the specification. In this section we reformulate their framework using our notation and prove its formal relationship with our characterization of AXps and CXps.

Let π be a trace, and let φ be a formula. A *candidate causal set* is a set $C \subseteq \mathsf{L}(\pi)$. Given $C \subseteq \mathsf{L}(\pi)$, the *alteration trace* π^C is a trace obtained by *replacing* every literal of π that occurs in C with their opposite. More formally, if $C \subseteq \mathsf{L}(\pi)$, then $\mathsf{L}(\pi^C) = (\mathsf{L}(\pi) \setminus C) \cup \overline{C}$. A *minimal cause for violation* is defined as follows:

Definition 4 (Minimal cause for violation)**.** Let π be a trace and φ a formula with $\pi \not\models \varphi$. A set $C \subseteq \mathsf{L}(\pi)$ is a minimal cause for violation of φ in counterexample π if $\pi^C \models \varphi$ and no proper subset $C' \subset C$ is such that $\pi^{C'} \models \varphi$. ■

Let $Alt_\pi(C) = \{\pi^X : X \subseteq C\}$ denote the set of alteration traces obtained by intervening on π over subsets of C. The following lemma establishes the correspondence between alteration and compatible traces.

Lemma 1. *Let π be a trace and let $E \subseteq \mathsf{L}(\pi)$. Then $Alt_\pi(\mathsf{L}(\pi) \setminus E) = \mathsf{Comp}_\pi(E)$.*

Proof. ($\rightarrow$) Let $\tau \in Alt_\pi(\mathsf{L}(\pi) \setminus E)$. Then there exists $X \subseteq \mathsf{L}(\pi) \setminus E$ such that $\tau = \pi^X$. By definition, $\mathsf{L}(\tau) = \mathsf{L}(\pi^X) = (\mathsf{L}(\pi) \setminus X) \cup \overline{X}$. Since $X \cap E = \emptyset$, every literal in E is left unchanged. Hence, $E \subseteq \mathsf{L}(\pi) \cap \mathsf{L}(\tau)$ that implies $\tau \in \mathsf{Comp}_\pi(E)$. ($\leftarrow$) Let $\tau \in \mathsf{Comp}_\pi(E)$. Consider the set $X = \mathsf{L}(\pi) \setminus \mathsf{L}(\tau)$. Since $E \subseteq \mathsf{L}(\pi) \cap \mathsf{L}(\tau)$ we have that $X \cap E = \emptyset$. Therefore, $X \subseteq \mathsf{L}(\pi) \setminus E$. Moreover, $\tau = \pi^X$ because the intervention flips exactly the literals on which π and τ differ. Hence, $\tau \in Alt_\pi(\mathsf{L}(\pi) \setminus E)$. □

Now we are ready to establish the relationships.

Proposition 4. *A set $C \subseteq \mathsf{L}(\pi)$ is a minimal cause for the violation of φ if and only if C is a CXp for (π, φ).*

Proof. ($\rightarrow$) Let C be a minimal cause for violation of the formula φ in the counterexample trace π. By definition, $\pi^C \models \varphi$. By Lemma 1, $Alt_\pi(C) = \mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus C)$. Since $\pi^C \in Alt_\pi(C)$ and $\pi^C \models \varphi$, we have that $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus C) \cap \mathsf{Mod}(\varphi) \neq \emptyset$. Hence, C is a WCXp. Suppose that C is not a CXp, then there exists a $C' \subset C$ that is also a WCXp. By Lemma 1 $Alt_\pi(C') = \mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus C')$, hence there exists $X \subseteq C'$ such that $\pi^X \models \varphi$. Since $X \subseteq C' \subset C$, this contradicts the minimality of C. ($\leftarrow$) Let C be a CXp (hence also a WCXp), so $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus C) \cap \mathsf{Mod}(\varphi) \neq \emptyset$. By Lemma 1, $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus C) = Alt_\pi(C)$, hence there exists $X \subseteq C$ with $\pi^X \models \varphi$. Take such an X that is subset-minimal (it exists, as $\mathsf{L}(\pi)$ is finite). Every $X' \subset X$ also satisfies $X' \subseteq C$, hence $\pi^{X'} \not\models \varphi$ by minimality of X; therefore X is a minimal cause for violation. By direction ($\rightarrow$), X is a CXp; since $X \subseteq C$ and C is a CXp, subset-minimality gives $X = C$. Therefore, C is a minimal cause. □

Unlike Meng and Alrajeh, our approach makes AXp/CXp duality explicit and enables efficient cause computation.

Complexity

We analyze the complexity of deciding whether a given set of temporal literals is a WAXp or WCXp. Trace length is fixed as part of the explanation setting and the decision problems amount to searching for models of LTL_f formulas over traces of bounded length. This reduces the complexity of satisfiability from PSPACE-complete to NP-complete.

Proposition 5. *Checking whether $E \subseteq \mathsf{L}(\pi)$ is a WCXp for an explanation problem (π, φ) is NP-complete.*

Proof (Sketch). [Membership.] A certificate is a trace $\tau \in (2^{\mathcal{A}})^{|\pi|}$. One can check in polynomial time that both $\tau \in \mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus E)$ and $\tau \models \varphi$. If both conditions hold, then $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus E) \cap \mathsf{Mod}(\varphi) \neq \emptyset$ and E is a WCXp.

[Hardness.] Hardness is shown by a reduction from the Boolean satisfiability problem (Garey and Johnson 1979). Let ϕ be a CNF formula over the variables $X = \{x_1, ..., x_n\}$,and let $q \notin X$ be a fresh variable. Set $\mathcal{A} = X \cup \{q\}$ and $|\pi| = 1$. We map ϕ to the instance (π, φ, E) where:

$$\varphi = q \wedge \phi, \qquad \pi(0) = \emptyset, \qquad E = \mathsf{L}(\pi).$$

Note that $\mathsf{L}(\pi) \setminus E = \emptyset$, so $\mathsf{Comp}_\pi(\emptyset)$ contains every trace of length 1 over $\mathcal{A}$; hence E is a WCXp iff some length-1 trace satisfies φ. The conjunct q ensures $\pi \not\models \varphi$, so (π, φ) is a well-formed explanation problem.

($\rightarrow$) Let M satisfy ϕ, and set $\tau(0) = \{x_i \mid M(x_i) = \text{true}\} \cup \{q\}$. Then $\tau \models q \wedge \phi$, so $\mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus E) \cap \mathsf{Mod}(\varphi) \neq \emptyset$ and E is a WCXp. ($\leftarrow$) Let E be a WCXp, witnessed by $\tau \in \mathsf{Comp}_\pi(\mathsf{L}(\pi) \setminus E) \cap \mathsf{Mod}(\varphi)$. Since $\varphi = q \wedge \phi$, we have $\tau \models \phi$; the assignment $M(x_i) = \text{true}$ iff $x_i \in \tau(0)$ satisfies ϕ. □

The complexity of WAXp checking follows by duality. $E \subseteq \mathsf{L}(\pi)$ is a WAXp for (π, φ) if and only if no trace in $\mathsf{Comp}_\pi(E)$ satisfies φ, that is, $\mathsf{L}(\pi) \setminus E$ is *not* a WCXp for (π, φ). WAXp checking is thus the complement of WCXp checking, and Proposition 5 yields the following.

Corollary 3. *Checking whether $E \subseteq \mathsf{L}(\pi)$ is a WAXp for an explanation problem (π, φ) is coNP-complete.*

Given the coNP (resp. NP) complexity of checking whether a set is a WAXp (resp. WCXp), a deletion-based procedure that removes literals one by one computes an AXp (resp. CXp) with a polynomial (linear) number of oracle calls, which matches general results for AXps/CXps (Marques-Silva and Ignatiev 2022).

Next, we show that checking whether a set E is an AXp (CXp) is DP-complete (coNP-complete) by reduction from MUS-checking (UNSAT) of a propositional formula. Let ϕ be an unsatisfiable propositional formula in 3-CNF with clauses $c_1, \ldots, c_m$, and let $R = \{r_1, \ldots, r_m\}$ be fresh propositional symbols, one per clause. Set $c'_i = r_i \vee c_i$ and $\phi' = \bigwedge_{i=1}^{m} c'_i$. Over the alphabet $\mathcal{A} = \text{vars}(\phi) \cup R$, the formula ϕ' is an LTL_f formula and π, with $\pi(0) = \emptyset$, is a trace of length one; since ϕ is unsatisfiable, $\pi \not\models \phi'$. Keeping r_i false forces c'_i to be satisfied through c_i; accordingly, we identify a set of clauses $S \subseteq \{c_1, \ldots, c_m\}$ with the temporal literals $\lambda(S) = \{\neg r_i : c_i \in S\} \subseteq \mathsf{L}(\pi)$. This yields the key correspondence: for every $S \subseteq \{c_1, \ldots, c_m\}$,

$$\mathsf{Comp}_\pi(\lambda(S)) \cap \mathsf{Mod}(\phi') \neq \emptyset \iff \textstyle\bigwedge_{c_i \in S} c_i \text{ is satisfiable.} (*)$$

Recall that $S \subseteq \phi$ is an MUS if S is unsatisfiable and every proper subset is satisfiable; S is a *maximal satisfiable subset* (MSS) if S is satisfiable and $S \cup \{c\}$ is unsatisfiable for every $c \in \phi \setminus S$; and S is an MCS if $\phi \setminus S$ is an MSS, i.e., a minimal set of clauses whose removal restores satisfiability.

Proposition 6. *S is an MUS of ϕ if and only if $\lambda(S)$ is an AXp for (π, ϕ').*

Proof. ($\rightarrow$) Let S be an MUS. By $(*)$, S unsatisfiable gives $\mathsf{Comp}_\pi(\lambda(S)) \cap \mathsf{Mod}(\phi') = \emptyset$, so $\lambda(S)$ is a WAXp; and for every $S' \subset S$, S' satisfiable gives, again by $(*)$, that $\lambda(S')$ is not a WAXp. These $\lambda(S')$ are exactly the proper subsets of $\lambda(S)$; hence $\lambda(S)$ is an AXp. ($\leftarrow$) Symmetric, reading each implication of $(*)$ in the opposite direction. □

Proposition 7. *Checking whether $E \subseteq \mathsf{L}(\pi)$ is an AXp for (π, φ) is DP-complete.*

Proof. Membership. E is an AXp iff (a) E is a WAXp and (b) no $E \setminus \{e\}$ is a WAXp. Condition (a) is in coNP (Corollary 3); (b) is in NP, as a certificate is, for some $e \in E$, a trace in $\mathsf{Comp}_\pi(E \setminus \{e\}) \cap \mathsf{Mod}(\varphi)$. Hence the problem is in DP. *Hardness.* By Proposition 6, since MUS-checking is DP-hard (MINIMAL UNSATISFIABILITY; (Papadimitriou and Wolfe 1988; Marques-Silva, Janota, and Mencía 2017)). □

Checking a CXp, on the other hand, is coNP-complete:

Proposition 8. *Checking whether $C \subseteq \mathsf{L}(\pi)$ is a CXp for (π, φ) is coNP-complete.*

Proof. [Membership.] By Proposition 4, C is a CXp if and only if (a) $\pi^C \models \varphi$ and (b) $\pi^{C'} \not\models \varphi$ for every $C' \subset C$. Condition (a) is a single model check on a fixed trace, feasible

Table 1: Per-instance AXp/CXp counts (top) and AXp/CXp explanation sizes in literals (bottom), by log. N: instances (C=complete, T=timeout). Column Px refers to the x-th quantile of the distributions, with P50 being the median. Max., min. refers to the maximum and minimum observed statistics.

	N		AXp										CXp									
			Complete					Timeout					Complete					Timeout				
Log	C	T	Min	P25	P50	P75	Max	Min	P25	P50	P75	Max	Min	P25	P50	P75	Max	Min	P25	P50	P75	Max
Per-instance counts																						
bpi2020-domdecl	81	8	1	10	21	82	441	567	578	588	647	834	1	1	3	5	19	18	23	28	35	70
bpi2020-intdecl	181	552	6	54	83	193	621	81	186	249	344	784	4	19	38	62	220	0	9	38	91	908
bpi2020-permlog	312	1166	4	40	177	443	1087	80	294	426	597	1253	2	15	51	143	366	0	9	70	223	885
bpi2020-prepcost	180	18	1	16	43	155	797	323	432	508	591	754	1	2	4	7	26	6	14	17	21	36
bpi2020-reqpay	79	1	1	6	12	62	937	829	829	829	829	829	1	1	2	4	26	88	88	88	88	88
sepsis	220	624	1	13	24	110	618	43	216	326	435	931	2	18	50	121	415	0	26	96	192	725
Explanation sizes																						
bpi2020-domdecl	81	8	1	3	4	4	5	1	4	4	5	6	1	10	14	16	24	18	22	24	30	38
bpi2020-intdecl	181	552	1	4	6	7	10	1	7	8	10	29	2	10	14	16	22	6	14	18	20	40
bpi2020-permlog	312	1166	1	5	6	7	11	1	5	6	8	55	2	18	20	20	24	16	24	26	30	144
bpi2020-prepcost	180	18	1	3	4	4	6	1	4	4	4	6	1	14	18	20	30	20	24	24	26	36
bpi2020-reqpay	79	1	1	3	4	4	6	1	4	4	4	7	1	10	16	20	24	24	26	26	28	30
sepsis	220	624	1	5	6	7	11	1	7	9	10	54	1	10	12	14	22	4	14	16	18	50

in polynomial time (Fionda and Greco 2018). Condition (b) is in coNP since a counterexample is a set $C' \subset C$ with $\pi^{C'} \models \varphi$, that can be checked in polynomial time.

[Hardness.] We reduce from UNSAT. Let ϕ be a CNF formula over $X = \{x_1, \ldots, x_k\}$. We assume w.l.o.g. that the all-true assignment does not satisfy ϕ, this can be checked in polynomial time, and if it does, we output a fixed instance that is not a CXp (e.g. any (π, φ) with $\pi \models \varphi$). Set $\mathcal{A} = X \cup \{q\}$ with q fresh, $|\pi| = 1$, $\pi(0) = \emptyset$, and $\psi = q \wedge \big(\neg(x_1 \wedge \cdots \wedge x_k) \rightarrow \phi\big)$. Since q is false in π, $\pi \not\models \psi$, so (π, ψ) is a well-formed explanation problem. Let $C = \mathsf{L}(\pi)$, we claim C is a CXp for (π, ψ) iff ϕ is unsatisfiable. Since π^C sets every atom to true, $\pi^C \models \psi$: q holds and the antecedent $\neg(\bigwedge_i x_i)$ is false. By Proposition 4, C is a CXp iff no proper subset $X' \subset C$ satisfies $\pi^{X'} \models \psi$; we show this holds iff ϕ is unsatisfiable.

($\rightarrow$) Suppose ϕ is unsatisfiable. Take any $X' \subset C$. If $q \notin X'$, then q stays false and $\pi^{X'} \not\models \psi$. If $q \in X'$, then since X' is proper some x_i stays false, so the antecedent $\neg(\bigwedge_i x_i)$ holds and $\pi^{X'} \models \psi$ would require $\pi^{X'} \models \phi$; but ϕ is unsatisfiable, so no such X' exists. Hence no proper subset satisfies ψ, and C is a CXp.

($\leftarrow$) Suppose ϕ is satisfiable. By the w.l.o.g. assumption, it has a model M different from all-true. Let $X' = \{q\} \cup \{x_i : M(x_i) = true\}$. Then $X' \subset C$, and $\pi^{X'} \models \psi$. Thus a proper subset satisfies ψ, and C is not a CXp. □

Computing AXps and CXps in ASP

We build upon the approach of (Fionda, Ielo, and Ricca 2026), that describes an Answer Set Programming (ASP) logic program $\Pi^k(\varphi)$, such that its stable models correspond one-to-one to traces π such that $|\pi| \leq k$ and $\pi \models \varphi$. $\Pi^k(\varphi)$ is a typical guess-and-check ASP encoding, with a *guess* program $\Pi^k_G(\varphi)$ that generates candidate traces, and a *check* program $\Pi^k_C(\varphi)$ that discards traces that are not models of φ. More in detail, $\Pi^k_C(\varphi)$ is a stratified logic program that evaluates the syntactic tree of φ in a bottom-up fashion over a guessed trace in $\Pi^k_G(\mathcal{A})$, discarding answer sets that encode traces that are not models of φ.

We briefly describe the original encoding for bounded LTL_f satisfiability.

Formulae & traces Consider a formula φ, and suppose each node in its syntax tree is assigned a unique integer. We can reify φ into ASP facts by means of predicates $until/3$, $negate/2$, $conjunction/2$, $atomic/2$, $next/2$, $true/1$, where the first term always refers to the unique identifier of the subformula and the other terms refer to identifiers of the most immediate subformulae (children). Predicate names are self-explanatory, and correspond to the given temporal/propositional operator.

Wlog, the root of φ is identified by 0. We denote this set of facts as $[\varphi]$. For example, the formula $\neg a \mathsf{U}(\mathsf{X}b)$ would be encoded as: `until(0,1,2). negate(1,3). atomic(3,a). next(2,4). atomic(4,b). root(0).`

A trace π can be encoded by means of facts $trace(t, a)$ if $a \in \sigma_t$, and $time(t)$ if $t < |\pi|$, where $trace(t, a)$ holds if $a \in \pi(t)$, and $time(i)$ if $|\pi| = k$ and $0 \leq i \leq k-1$. For example, the trace $\{a, b\} \cdot \{\} \cdot \{c\}$ is encoded as: `time(0..2). trace(0,a). trace(0,b). trace(2,c).` We denote this set of facts as $[\pi]$.

Semantics of temporal operators The ASP encoding of temporal operators makes use of predicates with signature $holds/2$, $trace/2$, and $time/1$. An atom $holds(t, x)$ models that the subformula rooted in x is true at time t. The atom $time(i)$ states that the trace has an i-th state, i.e. it has length at least i. The following logic program Π_{LTLf} models the semantics of temporal operators:

```
holds(T,X) :- atomic(X,A), trace(T,A).
holds(T,X) :- next(X,F), holds(T+1,F).
holds(T,X) :- conjunction(X,_), time(T),
    holds(T,F): conjunction(X,F).
holds(T,X) :- negate(X,F), time(T), not holds(T,F).
holds(T,X) :- until(X,F,G), holds(T,G).
holds(T,X) :- until(X,F,G), holds(T,F), holds(T+1,X).
```

```
sat :- holds(0,X), root(X).
```

Each rule closely mirrors the definition of the *next normal form* transformation of each temporal operator, so correctness follows from previous results (Fionda and Greco 2018); since this is a locally stratified logic program (Gebser et al. 2012), it admits a unique answer set containing sat iff $\pi \models \varphi$.

Search for a model The above encoding evaluates the formula φ over a trace π, so it is sufficient to replace the set of facts $[\pi]$ by some choice rules that generates all possible traces to lift it for checking bounded satisfiability. Consider the following logic program Π^k_{Search}:

```
symbol(X) :- atomic(_,X).
{ last_instant(T): T=0..k-1 } = 1.
time(0..T) :- last_instant(T).
{ trace(T,A): symbol(A) } :- time(T).
:- not sat.
```

Above, `k` is a runtime CLINGO constant which denotes the search bounds. The first choice rule yields the length of the guessed trace, $last_instant/1$, whereas the second one generates all possible states (given symbols that appear in φ), for each time point, thus all possible traces of length up to k. The constraint discards traces which are not models of φ. If the above logic program is satisfiable, projecting answer sets on predicates $trace/2$, $time/1$ allows one to obtain satisfying traces.

Thus, for the purposes of our paper, we have that $\Pi^k_G(\varphi) = \Pi^k_{Search}$, and $\Pi^k_C(\varphi) = [\varphi] \cup \Pi_{LTLf}$.

To apply our approach, we require to edit on $\Pi^k_G(\varphi)$, so that we fix the input trace π as the unique model of the guess program. This is done as follows.

Let $\pi \in (2^{\mathcal{A}})^t$. The program $G(\pi)$ containing a pair of rules of the form:

```
trace(t,a) :- not ntrace(t,a).
ntrace(t,a) :- not trace(t,a).
```

for each $(a,t) \in \mathcal{A} \times \{0,\ldots,|\pi|-1\}$, guesses the possible interventions on an input trace π, that is an arbitrary trace of length k by Lemma 1 with $E = \emptyset$.

Let's introduce the set of constraints $C(S) = \{$`:- not trace(t,a)` $: a_t \in S\} \cup \{$`:- not ntrace(t,a)` $: \neg a_t \in S\}$. Intuitively, these constraints *lock* certain events in the models of $G(\pi)$, playing the same role as $\mathsf{Comp}_\pi(\cdot)$.

Proposition 9. *Let π be a trace over $\mathcal{A}$, $S \subseteq \mathsf{L}(\pi)$. Models of $G(\pi) \cup C(S)$ are in one-to-one correspondence with $\mathsf{Comp}_\pi(S)$.*

Proof (Sketch). By construction observing that $G(\pi)$ generates $(2^{\mathcal{A}})^t$ and constraints in $C(S)$ force the occurrence/absence of the temporal literals $\ell_t \in S$. □

The logic program $\Pi^k_C(\varphi)$ of the original encoding of (Fionda, Ielo, and Ricca 2024) models the satisfaction of φ, thus:

Proposition 10. *Answer sets of $G(\pi) \cup C(S) \cup \Pi^k_C(\varphi)$ are in one-to-one correspondence with traces in $\mathsf{Comp}_\pi(S) \cap \mathsf{Mod}(\varphi)$.*

Note also that whenever $S = \mathsf{L}(\pi)$, the program $G(\pi) \cup C(S)$ admits an unique model, that is the trace π. Thus, if $\pi \not\models \varphi$, we have that $G(\pi) \cup C(\mathsf{L}(\pi)) \cup \Pi^k_C(\varphi)$ is incoherent. Consequently, it admits unsatisfiable cores and correction sets (Alviano et al. 2023). In particular, we are interested in MUSes and MCSes in terms of the constraints in $C(S)$.

Let $O = \{o(t,a) : a \in \mathcal{A}, 0 \leq t < |\pi|\}$ be a set of fresh objective atoms. Let $C^*(S) = \{$`:- o(t,a), not trace(t,a)` $: a_t \in S\} \cup \{$`:- o(t,a), not ntrace(t,a)` $: \neg a_t \in S\}$. Intuitively, each $o(t,a) \in O$ acts as a *selector* to include/exclude a constraint in $C(S)$, effectively *locking* certain temporal literals to assume the same value as in π.

Proposition 11. *Let $U \subseteq O$. Let $\Pi = G(\pi) \cup \Pi^k_C(\varphi) \cup C^*(\mathsf{L}(\pi))$. We have that $enforce(\Pi, O, U)$ is incoherent if and only if $S = \{a_t : o(t,a) \in U, a \in \pi(t)\} \cup \{\neg a_t : o(t,a) \in U, a \notin \pi(t)\}$ is a WAXp.*

Proof (Sketch). Follows from observing that including or excluding objective atoms in U respectively forces or forbids the corresponding temporal literals to match π. □

Notice this also implies the results holds for MUSes and AXps, since those are minimal unsatisfiable cores and minimal WAXps. The analogous for MCSes follows trivially from minimal hitting set dualities between MUSes/MCSes in ASP programs (Alviano et al. 2023), and AXps/CXps (Ignatiev et al. 2020):

Proposition 12. *Let $U \subseteq O$. Let $\Pi = G(\pi) \cup \Pi^k_C(\varphi) \cup C^*(\mathsf{L}(\pi))$. We have that $enforce(\Pi, O, O \setminus U)$ is coherent if and only if $S = \{a_t : o(t,a) \in U, a \in \pi(t)\} \cup \{\neg a_t : o(t,a) \in U, a \notin \pi(t)\}$ is a WCXp.*

Experiments

We evaluate the effectiveness of MUSes/MCSes enumeration for computing AXps and CXps of LTL_f specifications.

Implementation. Our implementation is a Python script that parses explanation problems of the form (π, φ) into a logic program, and then applies the MARCO (Liffiton et al. 2016) algorithm for ASP (Alviano et al. 2023), to jointly enumerate MUSes (AXps) and MCSes (CXps). We use the `clingo` (Gebser et al. 2019) ASP solver.

Execution Environment. Experiments were run on a machine equipped with a 13th Gen Intel(R) Core(TM) i7-13700 CPU and 32 GB of RAM, with 16 physical cores and 24 logical CPUs, running Ubuntu 22.04. In the experiments, we set a CPU-time limit of 10 seconds, a wall-clock timeout of 20 seconds, and a memory limit of 2 GB of RAM.

Data. Our benchmark is built from *Declare* (van der Aalst, Pesic, and Schonenberg 2009), a declarative process modeling language based on a set of templates whose semantics is given in LTL_f. In this setting, process discovery algorithms learn Declare models—that one can translate in LTL_f formulas—from event logs, i.e., multi-sets of traces recording real process executions. We choose Declare because it is a major application of LTL_f and, to the best of our knowledge, the only domain providing both real-world traces and LTL_f formulas that are learned from data rather than hand-crafted. We consider a collection of 6 real-world event logs (from the BPI Challenge 2020 (van Dongen 2020), and the Sepsis

event log (Mannhardt and Blinde 2017)). For each event log, we learn a Declare model using the Declare4Py library (Donadello et al. 2022), which we translate into an LTL_f formula φ. For each *variant* π (i.e., unique trace) in the event log, we generate one explanation problem: (π, φ) when $\pi \not\models \varphi$, and $(\pi, \neg\varphi)$ when $\pi \models \varphi$. We obtain a total of 3467 explanation problems, and we discard 45 where the corresponding LTL_f formula is incoherent–which would admit a single (empty) AXp and no CXps. This yields a total of 3422 instances.

Results. Table 1 reports statistics on the number, size of AXps and CXps found by our approach, partitioned by corresponding event log. Top-half reports the number of AXps/CXps found; bottom-half their size. We distinguish between *complete* instances, where enumeration terminates and *incomplete* instances where it does not terminate[1]. The approach is effective, as in a short timeout we are able to complete about 30% of the instances, while on the remaining *incomplete* instances we are still able to output several CXps and AXps, with a few exceptions (155) where we find zero CXps. However, this is due to the fact that the enumeration algorithm jointly enumerates CXps and AXps, and those instances contain a large amount of AXps. Additionally, we report no memory out errors, and an average (median) time-to-first AXp of 0.091 ± 0.032s (0.09s) and an average time-to-first CXp of 1.05 ± 1.58s (0.41s).

Comparison with Meng and Alrajeh (2025). The only existing tool that can compute our CXps is `caupybara` (Meng and Alrajeh 2025), which computes causes (equivalent to CXps) by brute-force. Candidate sets of increasing size (up to a user-provided upper bound on size) are enumerated, and checked for being a cause. As one may notice in Table 1, CXps of large size are not unlikely in real-world event logs. Thus, it is expected the approach scales poorly, as the size of causes increases. We highlight this limitation with a synthetic formula. Let $p(i) = \mathsf{G}\,(a_i \rightarrow \mathsf{X}_w\ b_i)$, $P(N) = \bigwedge_{i=1}^{N} p(i)$, and $\tau_N = \{a_1\}\{a_2\}\cdots\{a_N\}\{\}$, where the final empty state ensures the last weak next is not trivially satisfied. Every conjunct $p(i)$ is violated, so $\tau_N \not\models P(N)$, and the instance has exactly N AXps of size 2 (each pairing the presence of a_i with the absence of b_i) and, by hitting-set duality, 2^N CXps of size N. Altering the trace to satisfy $P(N)$ must satisfy each $p(i)$, choosing for each i whether to remove a_i or add b_i. Thus, one needs to enumerate up to N. Note that the conjunct $p(i)$ is not a "pathological" formula, but a rather common specification pattern in LTL_f specifications ("every time a_i happens, either the trace concludes or b_i happens next"). In particular, this is known as *Chain Response* in Declare (with a strong next replacing the weak).

Figure 2 reports the time to compute one and all CXps as N grows, under the same 10-second budget as before, with `caupybara` run at $bound = N$. `caupybara` exceeds the budget beyond N=4, *even for a single CXp*. Our approach, instead, manages to enumerate all CXps up to N=12, while also computing AXps. A CXp-only variant of our approach (enumerating *minimal stable models* (Alviano et al. 2023) to obtain MCSes/CXps, rather than applying MARCO-style joint MUS/MCS enumeration) matches `caupybara`'s task exactly, and finds a first CXp in under 0.1s, but performs worse in enumeration, completing only up to N=10, overall slowing down in the CXp enumeration task.

[1]MARCO is an anytime algorithm, outputs AXps/CXps as soon as they are found, and terminates when *all* CXps/AXps are found.

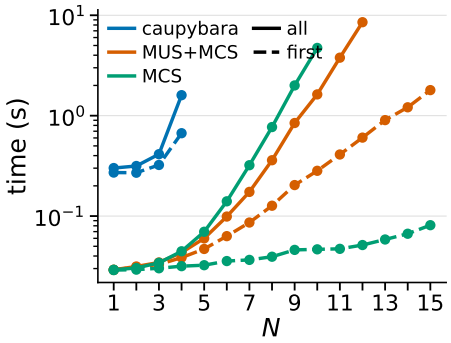


Figure 2: Time to the first and to all CXps on $P(N)$, τ_N, for `caupybara`, our full approach, and a CXp-only variant. Dashed: first CXp; solid: all CXps. 10 seconds timeout.

Conclusion

In this paper, we specialize abductive and contrastive explanations to LTL_f. We provide an idiomatic definition that side-steps the classifier view the original definitions require, prove the two equivalent, and prove contrastive explanations equivalent to a recently-proposed notion of temporal causality. This characterization lets us enumerate explanations efficiently by exploiting MUS-MCS duality, which we instantiate in Answer Set Programming. Experiments show the approach is effective on real-world event logs.

References

Alviano, M.; Dodaro, C.; Fiorentino, S.; Previti, A.; and Ricca, F. 2023. ASP and subset minimality: Enumeration, cautious reasoning and MUSes. *Artif. Intell.*, 320: 103931.

Bacchus, F.; and Kabanza, F. 1998. Planning for temporally extended goals. *Annals of Mathematics and Artificial Intelligence*, 22: 5–27.

Beer, I.; Ben-David, S.; Chockler, H.; Orni, A.; and Trefler, R. J. 2009. Explaining Counterexamples Using Causality. In *CAV*, volume 5643 of *Lecture Notes in Computer Science*, 94–108. Springer.

Brewka, G.; Eiter, T.; and Truszczynski, M. 2011. Answer set programming at a glance. *Commun. ACM*, 54(12): 92–103.

Chiariello, F.; and Marques-Silva, J. 2025. Formal Explanations of Black-Box Ranking Functions. In *JELIA (1)*, volume 16093 of *Lecture Notes in Computer Science*, 28–44. Springer.

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

De Giacomo, G.; De Masellis, R.; Grasso, M.; Maggi, F. M.; and Montali, M. 2014. Monitoring business metaconstraints based on LTL and LDL for finite traces. In *Business*

Process Management: 12th International Conference, 1–17. Springer.

De Giacomo, G.; Iocchi, L.; Favorito, M.; and Patrizi, F. 2019. Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications. In *Proceedings of the international conference on automated planning and scheduling*, volume 29, 128–136.

De Giacomo, G.; and Rubin, S. 2018. Automata-Theoretic Foundations of FOND Planning for LTLf and LDLf Goals. In *Proc. of IJCAI*, 4729–4735.

De Giacomo, G.; and Vardi, M. Y. 2013. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *IJCAI*, 854–860.

Donadello, I.; Riva, F.; Maggi, F. M.; and Shikhizada, A. 2022. Declare4Py: A Python Library for Declarative Process Mining. *BPM (PhD/Demos)*, 3216: 117–121.

Ferrère, T.; Maler, O.; and Ničković, D. 2015. Trace diagnostics using temporal implicants. In *International Symposium on Automated Technology for Verification and Analysis*, 241–258. Springer.

Fionda, V.; and Greco, G. 2018. LTL on Finite and Process Traces: Complexity Results and a Practical Reasoner. *J. Artif. Intell. Res.*, 63: 557–623.

Fionda, V.; Ielo, A.; and Ricca, F. 2024. LTLf2ASP: LTLf Bounded Satisfiability in ASP. In *LPNMR*, volume 15245 of *Lecture Notes in Computer Science*, 373–386. Springer.

Fionda, V.; Ielo, A.; and Ricca, F. 2026. Computing Syntax Tree-based Minimal Unsatisfiable Cores of LTLf Formulas. In *AAAI*, 19073–19081. AAAI Press.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2012. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2019. Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, 19(1): 27–82.

Gelfond, M.; and Lifschitz, V. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Gener. Comput.*, 9(3/4): 365–386.

Granada, J. C.; Ignatiev, A.; and Stuckey, P. J. 2026. A Formal Framework for the Explanation of Finite Automata Decisions. *CoRR*, abs/2602.13351.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2019. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5): 93:1–93:42.

Halpern, J. Y. 2016. *Actual Causality*. MIT Press.

Ielo, A.; Mazzotta, G.; Peñaloza, R.; and Ricca, F. 2026. Enumerating Minimal Unsatisfiable Cores of LTLf Formulae. In *AAAI*, 19160–19168. AAAI Press.

Ignatiev, A.; Narodytska, N.; Asher, N.; and Marques-Silva, J. 2020. From Contrastive to Abductive Explanations and Back Again. In *AI*IA*, volume 12414 of *Lecture Notes in Computer Science*, 335–355. Springer.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019. Abduction-Based Explanations for Machine Learning Models. In *AAAI*, 1511–1519. AAAI Press.

Kress-Gazit, H.; Fainekos, G. E.; and Pappas, G. J. 2009. Temporal-logic-based reactive mission and motion planning. *IEEE transactions on robotics*, 25(6): 1370–1381.

Liffiton, M. H.; Previti, A.; Malik, A.; and Marques-Silva, J. 2016. Fast, flexible MUS enumeration. *Constraints An Int. J.*, 21(2): 223–250.

Liffiton, M. H.; and Sakallah, K. A. 2008. Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints. *J. Autom. Reason.*, 40(1): 1–33.

Lifschitz, V. 2019. *Answer Set Programming*. Springer.

Mannhardt, F.; and Blinde, D. 2017. Analyzing the Trajectories of Patients with Sepsis using Process Mining. In *RADAR+EMISA@CAiSE*, volume 1859 of *CEUR Workshop Proceedings*, 72–80. CEUR-WS.org.

Marques-Silva, J.; and Ignatiev, A. 2022. Delivering Trustworthy AI through Formal XAI. In *AAAI*, 12342–12350. AAAI Press.

Marques-Silva, J.; Janota, M.; and Mencía, C. 2017. Minimal sets on propositional formulae. Problems and reductions. *Artif. Intell.*, 252: 22–50.

Meng, H.; and Alrajeh, D. 2025. Causal Detection in Linear Temporal Counterexamples. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 1418–1423.

Niu, T.; Xiao, S.; Zhang, X.; Li, J.; Huang, Y.; and Shi, J. 2024. Computing minimal unsatisfiable core for LTL over finite traces. *J. Log. Comput.*, 34(7): 1274–1294.

Papadimitriou, C. H.; and Wolfe, D. 1988. The Complexity of Facets Resolved. *J. Comput. Syst. Sci.*, 37(1): 2–13.

Pnueli, A. 1977. The Temporal Logic of Programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, 46–57. IEEE Computer Society.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial intelligence*, 32(1): 57–95.

Roveri, M.; Ciccio, C. D.; Francescomarino, C. D.; and Ghidini, C. 2024. Computing Unsatisfiable Cores for LTLf Specifications. *J. Artif. Intell. Res.*, 80: 517–558.

Shih, A.; Choi, A.; and Darwiche, A. 2018. A Symbolic Approach to Explaining Bayesian Network Classifiers. In *IJCAI*, 5103–5111. ijcai.org.

van der Aalst, W. M. P.; Pesic, M.; and Schonenberg, H. 2009. Declarative workflows: Balancing between flexibility and support. *Comput. Sci. Res. Dev.*, 23(2): 99–113.

van Dongen, B. F. 2020. BPI Challenge 2020. Dataset.