FROM REASONS TO REVISIONS: A THEORY OF INTERACTIVE BELIEF CHANGE FOR BOOLEAN CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Explanations of learned classifiers are typically one-directional: a model justifies a prediction, but disagreement with that explanation does not provide a principled way to revise the model. We propose a human-in-the-loop framework in which explanations can be contested and used to correct classifier behaviour. Grounded in sufficient reasons for Boolean classifiers, correction is formulated as a minimal adjustment of the reasons supporting a prediction, connecting explainable machine learning with belief change. We characterise which classifier edits can be obtained by adjusting reasons and how restrictions on these adjustments capture a formal notion of minimal change. The main computational challenge is that a prediction may admit exponentially many sufficient reasons, making explicit enumeration impractical. For classifiers compiled as sentential decision diagrams (SDDs), we restrict the search to minimum-disagreement reasons using nearest-model enumeration and residual reduction. We assume that the reasons most relevant for user correction are those that disagree minimally with the contested instance. This avoids materialising the complete set of prime implicants before identifying the minimal corrections, although complete recovery of tied reasons remains exponential in the worst case. Experiments demonstrate the practical advantage over exhaustive prime-implicant enumeration on the tested classifiers. Since learned models such as decision trees can be compiled into these representations, our approach provides a symbolic correction layer for learned predictors and supports an interactive neuro-symbolic explanation-revision loop.

1 INTRODUCTION

Explaining the decisions of classification models has received considerable attention, leading to methods such as feature attribution Lundberg & Lee (2017); Ribeiro et al. (2016), counterfactual explanations Verma et al. (2024), and logic-based approaches Darwiche (2023); Marques-Silva (2024). These methods typically present explanations through a one-way interaction, from the model to the user. The core idea of this paper is that, if we have a human-understandable explanation for a decision, we can also scrutinise and correct the explanation if the decision was wrong. Moreover, by correcting the explanations, we provide the information needed to modify the classifier itself to avoid similar errors in the future. Our research question is therefore: *How can user-provided corrections to explanations be incorporated into a classifier in a principled manner?*

Our approach builds on two developments in logic-based explainability. The first is the notion of *sufficient reasons* Darwiche & Hirth (2020), also studied as PI explanations Shih et al. (2018) and abductive explanations Ignatiev et al. (2019). A Boolean classifier is represented by a propositional formula Δ, and a sufficient reason is a prime implicant of Δ: a minimal set of feature conditions sufficient for the classification. The second development is the framework of *edit operators* for Boolean classifiers introduced by Schwind et al. (2023), which studies classifier modification from the perspective of belief change.

We connect these two ideas by defining *interactive edit operators*: classifier edits induced by correcting sufficient reasons. Rather than specifying only that an instance should change classification, the user may resolve choices about *why* the original decision was wrong, selecting which sufficient reason to modify and how it should be weakened. The resulting correction is then applied directly to

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.

the symbolic classifier. This allows classifier revision to operate at the level of the reasons supporting a decision, rather than only at the level of its output.

This interaction also distinguishes our proposal from classical belief revision approaches, which typically resolve alternative revisions using an ordering that represents which beliefs more entrenched Alchourrón et al. (1985); Gärdenfors (1988); Katsuno & Mendelzon (1992). We do not assume such an ordering: when several minimal corrections are possible, the choice is delegated to the user. This is particularly natural for classifier correction, where the intended generalisation of a fix may depend on domain knowledge not part of the model itself. Related work on Interactive XAI Rago & Martinez (2024) similarly uses user feedback on explanations, but maintains an explanation knowledge base separate from the classifier. Our operators instead modify the symbolic classifier itself.

Our main theoretical contribution is two representation theorems. The first characterises positive unary edit operators through basic constraints on reason adjustment; the second gives corresponding constraints for the operators satisfying our postulates for minimal change. Minimal weakening then provides a concrete family of adjustments within this characterised class.

A practical difficulty is that sufficient reasons are prime implicants, of which a classifier may have exponentially many. Our second goal is therefore to optimise the search for the reasons relevant to an edit. The procedure operates on classifiers compiled as SDDs and exploits conditioning, residual support, necessary conditions, and counterexample-guided minimal-hitting-set enumeration to reduce the remaining search space. Although complete recovery of tied reasons can still be exponential in the reduced residual dimension, experiments show that this restricted enumeration is substantially faster than first materialising all prime implicants on the tested classifiers.

Although our formal development concerns Boolean classifiers, the framework is intended as a symbolic correction layer. Decision trees can be represented directly in such forms, while other learned classifiers may be compiled or approximated by symbolic representations for explanation Shih et al. (2018); Huang et al. (2022); Darwiche & Hirth (2023). Our guarantees apply to the compiled Boolean classifier; transferring an edit back to an arbitrary underlying learned model is a separate problem and is left for future work. We likewise focus here on the formal foundations of interaction rather than on a particular user interface. Figure 1 illustrates one possible dialogue-based realisation, in which the model presents sufficient reasons for a decision, the user contests or adjusts them, and the resulting correction is translated into an edit of the classifier. Such an interface could guide the user through competing minimal corrections while keeping the underlying revision process explicit.

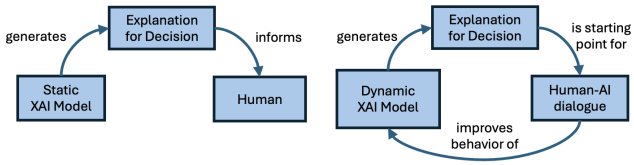


Figure 1: Static vs. Dynamic Explainable AI

The remainder of the paper introduces the formal background, defines interactive edits and minimal change of reasons, studies their properties, develops the SDD-based procedure, and evaluates its computational behaviour.

2 PRELIMINARIES

In this section we present notation and definitions that will be used throughout this paper. We use lowercase letters $(a, b, c, \ldots)$ to denote atoms. We assume that we work with a fixed finite set At of atoms. We denote by $\mathcal{L}(At)$ the propositional language that includes At and is closed under the usual logical connectives. A *literal* is an atom or the negation of an atom. We use l to denote arbitrary literals and $\bar{l}$ to denote the complement of l. A *term* is a conjunction of literals. We also treat terms as sets of literals so that $\tau \subseteq \tau'$ means that every literal that appears in τ also appears in

τ'. We assume throughout this paper that a term is consistent (i.e., contains no contrary literals). We define the empty term $\tau = \emptyset$ to be equivalent to the conjunctive identity $\top$. When denoting terms we sometimes omit conjunction symbols and use overlines for negation, such as $a\overline{bc}$ for the term $a \wedge \neg b \wedge \neg c$. Given a term τ we denote by $\overline{\tau}$ the literal-wise complement of τ, e.g., if $\tau = a \wedge \neg b$ then $\overline{\tau} = \neg a \wedge b$. We define a *valuation* of At to be a term in which every atom in At appears exactly once. We will use ω to denote an arbitrary valuation. We write $\omega \models \phi$ if ω is a model of ϕ in the usual sense. We denote by $[\phi]$ the models of ϕ and write $\phi \models \psi$ iff $[\phi] \subseteq [\psi]$.

Following Darwiche & Hirth (2020) we represent a Boolean classifier with input features At by a propositional formula $\Delta \in \mathcal{L}(At)$, called the *decision formula*. The input instances for this classifier are the valuations of At. A decision formula Δ classifies ω positively if $\omega \models \Delta$, and negatively if $\omega \not\models \Delta$. We use the notion of *sufficient reason* to explain classifier decisions Darwiche & Hirth (2020). Sufficient reasons are based on *prime implicants*. A term τ is an *implicant* of a formula ϕ if $\tau \models \phi$. A term τ is a *prime implicant* of a formula ϕ if τ is an implicant of ϕ and there is no $\tau' \subset \tau$ that is an implicant of ϕ. We denote by $PI(\phi)$ the set of prime implicants of ϕ. We say that a term τ *applies* to an instance ω if $\tau \subseteq \omega$. Following Darwiche & Hirth (2020) we define a *sufficient reason* for a decision $\omega \models \Delta$ (resp. $\omega \not\models \Delta$) to be a prime implicant of Δ (resp. $\neg\Delta$) that applies to ω. Without reference to an instance, we also refer to the prime implicants of Δ (resp. $\neg\Delta$) as *the positive (resp. negative) sufficient reasons of* Δ. Thus, a positive (resp. negative) sufficient reason is a minimal condition of *any* instance that guarantees a positive (resp. negative) classification, and a sufficient reason *for a decision* $\omega \models \Delta$ (resp. $\omega \not\models \Delta$) is a minimal condition *of the instance* ω that guarantees a positive (resp. negative) classification. If $\Delta \models \tau$ (resp. $\neg\Delta \models \tau$) then τ is a *necessary* positive (resp. negative) reason. A term τ is a necessary positive (resp. negative) reason for Δ if and only if τ is a subset of all the sufficient positive (resp. negative) reasons for Δ Darwiche & Hirth (2020). The intersection of a nonempty set of positive or negative sufficient reasons may be empty, in which case the only corresponding necessary reason is the empty term $\top$.

Let us look at an example. The classifier

$$\Delta = (m \rightarrow r) \wedge (f \rightarrow w) \tag{1}$$

encodes a dining preference: if we eat meat (m) we drink red wine (r) and if we eat fish (f) we drink white wine (w). The positive sufficient reasons for Δ are:

$$\neg m \wedge \neg f, \neg m \wedge w, r \wedge \neg f, r \wedge w. \tag{2}$$

The negative sufficient reasons for Δ are:

$$m \wedge \neg r, f \wedge \neg w \tag{3}$$

The instance $\omega = mr\overline{fw}$ (meat with red wine) is classified as positive: $\omega \models \Delta$. The unique sufficient reason for this decision is $r \wedge \neg f$. The instance $\omega = m\overline{r}\overline{f}w$ (meat with white wine) is classified negatively: $\omega \not\models \Delta$. The sufficient reason for this decision is $m \wedge \neg r$. The decision formula Δ has no nonempty necessary reasons.

3 Classifier Revision by Adjustment of Reasons

Before formalising our approach, we begin with a sketch of the main idea. Consider the decision formula Δ as defined in equation (1) and consider again the instance $\omega = mr\overline{f}w$, which is classified negatively: $\omega \not\models \Delta$. Suppose we want to modify Δ so that it classifies ω positively. As it happens, (1) is not a large or complex formula so we can modify it manually. For instance, we can weaken the implication $m \rightarrow r$ by strengthening its antecedent to $m \wedge f$ or by weakening its consequent to $r \vee w$, leading to

$$\Delta_1 \equiv ((m \wedge f) \rightarrow r) \wedge (f \rightarrow w),$$
$$\Delta_2 \equiv (m \rightarrow (r \vee w)) \wedge (f \rightarrow w).$$

Both Δ_1 and Δ_2 classify ω as positive. The problem is that, in practice, Δ is often a large and complex formula that cannot easily be understood. Moreover, Δ may be represented not with a propositional formula but with a less interpretable structure such as a Boolean circuit Darwiche & Hirth (2023). Thus, modifying the decision formula directly can be difficult.

Instead of modifying Δ directly, we model this modification as a process of *adjusting its sufficient reasons*. This approach allows for human-guided modification of a classifier, by correcting the explanations behind decisions we wish to change. Consider the sufficient reasons for Δ (see (2)). Since

Δ classifies ω as negative, none of these reasons apply to ω. However, we can weaken one or more of the reasons so that they do. For example, the reason $\neg m \wedge \neg f$ implies that we accept everything as long as we eat neither meat nor fish. Removing $\neg m$ yields $\neg f$, i.e., we accept everything as long as we do not eat fish. The new classifier is then formed by the disjunction of this weakened reason and the other reasons: $(\neg f) \vee (\neg m \wedge w) \vee (r \wedge \neg f) \vee (r \wedge w)$. This formula is equivalent to Δ_1. Alternatively, if we allow white wine even with meat, we can weaken $\neg m \wedge w$ by removing $\neg m$. If we similarly combine this weakened reason with the others, we obtain a formula that is equivalent to Δ_2. This example highlights the value of human input: the choice between weakening one reason to obtain Δ_1 and another to obtain Δ_2 depends on the intended dining preference.

We formalise this process in stages. First, we recall positive unary edit (PE) operators and three simple examples. We then define adjustment functions and show that two basic constraints on these functions characterise all PE operators through a representation theorem. Additional constraints capture a particular notion of minimal change of reasons, and a second representation theorem identifies the corresponding class of PE operators.

Finally, we introduce *minimal weakening* adjustment functions as a concrete special case. The number of sufficient reasons can be exponential in the number of variables. Minimal weakening restricts the candidate reasons to those that *disagree least* with the target instance. Although this candidate set can still be exponential in the worst case, it is typically much smaller than the full set of prime implicants. Moreover, in Section 4 we develop a procedure for recovering these candidates without having to enumerate all prime implicants. Proofs of the representation theorems and the minimal weakening result are given in Appendix A.1.

3.1 Positive Unary Edit Operators

Edit operators were introduced by Schwind et al. (2023) to study the modification of Boolean classifiers from the perspective of belief change. They define an edit operator as a function $\diamond$ that maps every decision formula Δ and every set of instances, represented by a formula ϕ, to a modified decision formula $\Delta' = \Delta \diamond \phi$. They propose several postulates and constructive definitions for such operators. We will restrict our attention to edit operators that take a single instance as input and refer to such operators as *unary edit operators*.

Definition 1. *A* unary edit operator *is a function $\oplus$ that maps every decision formula Δ and instance ω to a new decision formula $\Delta' = \Delta \oplus \omega$.*

Positive edits require the target instance to be classified as positive after the edit. Inspired by the ***KM postulates*** for belief revision Katsuno & Mendelzon (1992), four *basic postulates* characterise the class of *positive edit* operators. The first three, adapted to the unary case, are:

(P1) $\omega \models (\Delta \oplus \omega)$.
(P2) If $\omega \models \Delta$ then $(\Delta \oplus \omega) \equiv \Delta$.
(P3) If $\Delta \equiv \Delta'$ then $(\Delta \oplus \omega) \equiv (\Delta' \oplus \omega)$.

(P1) states that the operator ensures that the given instance is afterwards classified as positive. (P2) relates to minimal change. It states that if the given instance is already classified as positive, then no changes are needed. (P3) reflects the principle of syntax-independence, i.e., the result of an edit depends only on the semantics of the decision formula. We do not impose their fourth postulate, which compares edits by different formulas rather than defining a condition on a single target instance[1].

Definition 2. *$\oplus$ is a* positive unary edit *(PE) operator if and only if it satisfies (P1), (P2) and (P3).*

There are three simple options among the set of PE operators Schwind et al. (2023). Their unary versions are defined as follows.

Definition 3. *The* trivial*,* basic *and* drastic *operators $\oplus_T$, $\oplus_B$ and $\oplus_D$ are defined by: $\Delta \oplus_T \omega = \Delta$ if $\omega \models \Delta$, otherwise $\Delta \oplus_T \omega = \top$; $\Delta \oplus_B \omega = \Delta \vee \omega$; and $\Delta \oplus_D \omega = \Delta$ if $\omega \models \Delta$, otherwise $\Delta \oplus_D \omega = \omega$.*

$\oplus_T$ leaves Δ unchanged if ω is already positive, and otherwise classifies *all* instances as positive. $\oplus_B$ simply adds ω to the set of positive instances. $\oplus_D$ leaves Δ unchanged if ω is already positive, and otherwise classifies *only* ω as positive. It is easy to see that all three are PE operators.

[1]This postulate relates to minimal change and states that, if $\psi \models \Delta \diamond \phi$ then $\Delta \diamond \phi \equiv \Delta \diamond (\phi \wedge \psi)$

Note that a *negative* edit operator can be characterised by negative versions of (P1)–(P3). Moreover, a negative operator $\ominus$ can be defined from a positive operator $\oplus$ by the identity $\Delta \ominus \omega \equiv \neg(\neg\Delta \oplus \omega)$ Schwind et al. (2023). We will focus in this paper only on positive edits.

3.2 Interactive Edit Operators

We now define a class of edit operators based on the *interactive* adjustment of reasons. These operators are defined by an *adjustment function* C which, given a set S of terms and a target instance ω, returns an adjusted set $C(S, \omega)$ of reasons. The idea is that this function abstracts from how the adjustments are elicited from the user, for example through a user interface or dialogue system.

Definition 4. *An* adjustment function C *maps every set S of consistent terms and every instance ω to a set $C(S, \omega)$ of consistent terms. It induces the unary edit operator*

$$\Delta \oplus_C \omega = \bigvee C(PI(\Delta), \omega).$$

An operator $\oplus$ is an interactive edit operator *iff there exists an adjustment function C such that $\Delta \oplus_C \omega \equiv \Delta \oplus \omega$ for every Δ and ω. In this case, we say that C* represents *$\oplus$.*

We assume that every adjustment function satisfies the following two conditions. Condition (C1) requires some output term to apply to the target instance ω, ensuring a positive classification. Condition (C2) requires the input to remain unchanged if some term already applies to ω. In belief change terminology, C1 and C2 correspond to *success* and *vacuity*.

(C1) $\tau \subseteq \omega$ for some $\tau \in C(S, \omega)$.
(C2) If $\tau \subseteq \omega$ for some $\tau \in S$ then $C(S, \omega) = S$.

These conditions can be thought of as two basic rules in the interactive editing process: the user must perform a relevant adjustment but only if needed. The following theorem shows that the class of PE operators and the class of interactive edit operators coincide. In other words, every interactive edit operator satisfies the three basic postulates from Section 3.1. Conversely, every operator satisfying the basic postulates can be represented as an interactive edit operator.

Theorem 1 (Representation of PE operators)**.** *$\oplus$ is an interactive edit operator if and only if it is a PE operator.*

The trivial, basic and drastic PE operators introduced above have simple representations as interactive edit operators:

Definition 5. *The adjustment functions C_T, C_B and C_D are defined by: $C_T(S, \omega) = C_B(S, \omega) = C_D(S, \omega) = S$, if $\tau \subseteq \omega$ for some $\tau \in S$; otherwise $C_T(S, \omega) = \{\top\}$, $C_B(S, \omega) = S \cup \{\omega\}$ and $C_D(S, \omega) = \{\omega\}$.*

Proposition 1. *The functions C_T, C_B and C_D satisfy (C1) and (C2) and represent $\oplus_T$, $\oplus_B$ and $\oplus_D$, respectively.*

3.3 Postulates for Minimal Change of Reasons

The basic postulates (P1), (P2) and (P3) are quite permissive, as they leave open the question of how an edit for a given target instance generalises to other instances. Indeed, the trivial, basic and drastic operators all satisfy the basic postulates. We now consider three additional postulates that express a kind of *minimal change*.

(PW) $\Delta \models \Delta \oplus \omega$.
(PR) If $\tau' \not\models \Delta$ and $\tau' \models \Delta \oplus \omega$ then $\exists \sigma \left(\sigma \subseteq \omega \cap \tau' \text{ and } \sigma \models \Delta \oplus \omega\right)$
(PC) $\Delta \models \tau$ and $\tau \subseteq \omega$ then $\Delta \oplus \omega \models \tau$.

Here τ' is an instance, and τ and σ are consistent terms. (PW) expresses preservation of previously positive instances (i.e., a positive edit weakens the decision formula). (PR) requires every newly accepted instance to share a sufficient condition with the target: the witness σ applies to both instances and guarantees acceptance by the edited classifier. Finally, (PC) ensures the preservation of every necessary reason that applies to the target.

Definition 6. *A PE operator is a* minimal change *operator iff it satisfies (PW), (PR) and (PC).*

The trivial and drastic operators are not minimal change operators. For example, let $\Delta = a \wedge b$ and $\omega = a \wedge \neg b$. The trivial operator returns $\top$, violating (PC): $\Delta \models a$ and $a \subseteq \omega$, but $\top \not\models a$. The drastic operator returns ω, violating (PW), since the previously positive instance $a \wedge b$ is no longer accepted. By contrast, it is not hard to prove that the basic operator satisfies all three additional postulates. The basic operator is in a sense the *most conservative* operator; it adds *only* the target instance and does not generalise to any other instances.

The corresponding requirements at the level of adjustment functions are the following constraints:

(CW) If $\tau \in S$ then $\tau' \subseteq \tau$ for some $\tau' \in C(S, \omega)$.
(CR) If $\tau \in C(S, \omega) \setminus S$ then $\tau \subseteq \omega$.
(CC) If $\tau \in C(S, \omega) \setminus S$, $\tau' \subseteq \omega$ and $\forall \tau'' \in S : \tau' \subseteq \tau''$ then $\tau' \subseteq \tau$.

Condition (CW) retains every original term or replaces it by a weaker term. Condition (CR) requires every introduced term to apply to the target. Condition (CC) requires introduced terms to retain any condition common to the original terms that also applies to the target. For nonempty S, its universal premise is equivalent to $\tau' \subseteq \bigcap S$; for $S = \emptyset$, this premise is vacuous. These constraints, together with (C1) and (C2), give the following complete characterisation.

Definition 7. *An adjustment function is* minimal *iff it satisfies (CW), (CR) and (CC).*

Theorem 2. *$\oplus$ is a minimal change operator iff it is represented by a minimal adjustment function.*

Theorems 1 and 2 provide two levels of representation: (C1)–(C2) characterise the PE operators, while adding (CW), (CR) and (CC) characterises those also satisfying (PW), (PR) and (PC). The second theorem describes all operators permitted by these constraints. The operators induced by the minimal weakening strategy in the following subsection form a subclass of the minimal change operators.

3.4 Minimal Weakening Adjustment Functions

We now give a concrete subclass of the adjustment functions characterised by Theorem 2. The strategy is to (1) select one or more reasons that *minimally disagree* with the reclassified instance, (2) weaken them by removing only the disagreeing literals, and (3) leave all other reasons unchanged. We call such adjustments *minimal weakening adjustments*, and show that they satisfy our postulates for minimal change of reasons. Here minimality additionally refers to the number of disagreeing literals removed from a selected reason. When several reasons tie for minimum disagreement, the user may select any nonempty subset of them. Restricting attention to these candidates may reduce the number of reasons to consider, although their number can still be exponential.

Given two terms τ, τ', let $dis_\tau(\tau')$ denote the *disagreement of τ with τ'*, i.e., the literals in τ that disagree with τ', defined by $dis_\tau(\tau') = \tau \cap \overline{\tau'}$. We define the total preorder $\leq_\omega^{dis}$ over terms by $\tau \leq_\omega^{dis} \tau'$ if and only if $|dis_\tau(\omega)| \leq |dis_{\tau'}(\omega)|$.

Definition 8. *A* minimal weakening *adjustment function is an adjustment function C with $C(\emptyset, \omega) = \{\omega\}$ and, for every nonempty S,*

$$C(S, \omega) = \{\tau \setminus dis_\tau(\omega) : \tau \in S_w\} \cup (S \setminus S_w),$$

where the selected set S_w satisfies $\emptyset \subset S_w \subseteq \min(S, \leq_\omega^{dis})$.

Proposition 2. *Every minimal weakening adjustment function C satisfies (C1), (C2), (CW), (CR) and (CC). Consequently, its induced operator satisfies (P1), (P2), (P3), (PW), (PR) and (PC).*

To give an example of minimal weakening, consider the following classifier. We eat no meat, drink white wine if we eat fish, and drink red or white wine.

$$\Delta = \neg m \wedge (f \rightarrow w) \wedge (w \vee r). \tag{4}$$

Consider the instance $\omega = \overline{mr}f\overline{w}$ which is classified negatively: $\omega \not\models \Delta$. Let us compute a minimal weakening adjustment of Δ. The initial positive reasons are $\neg m \wedge r \wedge \neg f$ and $\neg m \wedge w$, none of which apply to ω. We have that $\neg m \wedge r \wedge \neg f$ disagrees with ω on two variables, and that $\neg m \wedge w$ disagrees with ω on one variable. Minimal weakening means that we weaken one or more of the sufficient reasons that disagree minimally with ω. In this case there is just one. Thus we weaken the reason $\neg m \wedge w$ to $\neg m$ and leave $\neg m \wedge r \wedge \neg f$ intact, leading to

$$\Delta \oplus \omega \equiv \neg m \vee \neg m \wedge r \wedge \neg f \equiv \neg m.$$

From adjustment to computation. The representation theorems describe which edits are permitted, while minimal weakening specifies a way to choose the reasons to change. To implement this strategy, we must recover the candidate set $\min(PI(\Delta), \leq_{\omega}^{dis})$ from which the user selects S_w. If $\Delta \equiv \bot$, the definition directly returns $\{\omega\}$; otherwise, for a rejected target, the following section develops an enumeration procedure for these candidates on an SDD representation of the classifier. This connects the adjustment strategy to computation without requiring an explicit enumeration of all of $PI(\Delta)$.

4 Computation of Minimum-Disagreement Edits

The minimal weakening strategy of Definition 8 requires identifying the sufficient reasons of the classifier that have minimum disagreement with ω. These reasons are selected from the prime implicants $PI(\Delta)$, whose number may be exponential in the number of features (Darwiche & Marquis, 2002). Explicitly generating all prime implicants only to identify the minimum-disagreement ones can therefore be prohibitively expensive. Instead of generating the complete set $PI(\Delta)$, we develop a procedure that enumerates exactly the minimum-disagreement sufficient reasons needed for minimal weakening. It combines minimum-distance computation over the compiled representation, conditioning to reduce the residual search space, and counterexample-guided minimal-hitting-set enumeration. We analyse the resulting procedure both theoretically, through its worst-case time complexity, and empirically on random classifiers.

We assume that Δ is compiled as a sentential decision diagram (SDD) (Darwiche, 2011). An SDD represents a Boolean function relative to a *vtree*, a binary tree whose leaves correspond to variables. An internal SDD decision node has the form

$$\bigvee_i (p_i \wedge s_i),$$

where each pair (p_i, s_i) is called an *element*. The prime p_i ranges over the variables on the left side of the corresponding vtree node, while the sub s_i ranges over those on the right.

Two properties are particularly important for our procedure. First, *structured decomposability* ensures that the prime and sub of each element use disjoint sets of variables. Because of this, quantities such as Hamming distance can be computed independently on the two components and combined additively. Second, the primes of a decision node form a mutually exclusive and exhaustive partition of the left-side assignments. A decision node can therefore be treated as a case distinction: for a complete assignment, exactly one element is selected. SDDs are also convenient for the remaining stages of the procedure. Conditioning on fixed literals produces a residual circuit in which irrelevant branches and variables may disappear, while satisfiability, model extraction, and related queries can be performed directly on the compiled representation. Moreover, SDDs are directed acyclic graphs, so identical subfunctions can be shared rather than represented repeatedly. The procedure therefore exploits both the logical structure of the representation and any reduction in the residual circuit obtained after conditioning.

4.1 Minimum-Disagreement Reason Enumeration

Assume Δ is satisfiable and $\omega \not\models \Delta$, and let $d^\star = \min_{\mu \models \Delta} \mathrm{Ham}(\mu, \omega)$. The following proposition provides the basis for the enumeration strategy.

Proposition 3 (Nearest-model characterisation)**.** *For every $\tau \in PI(\Delta)$, $\tau \in \min(PI(\Delta), \leq_{\omega}^{dis})$ if and only if there exists a model $\mu \models \Delta$ such that $\mathrm{Ham}(\mu, \omega) = d^\star$ and $\tau \subseteq \mu$.*

By Proposition 3, the minimum-disagreement reasons are exactly the prime implicants contained in the nearest accepted instances. This allows us to search for the required reasons without first enumerating $PI(\Delta)$. The proof is given in Appendix A.2.

Algorithm 1, given in Appendix B, implements this idea in two stages. First, a bottom-up pass over the SDD computes $d^\star$. This is a target-relative form of minimum-cardinality computation (Darwiche, 2001) and takes $O(|D|)$ time for an SDD D. The accepted instances at that distance are then enumerated using the stored distances; their number can be exponential.

Second, for each nearest instance μ, Algorithm 2 recovers all prime implicants contained in μ. Before enumeration, it fixes the literals on which μ differs from ω, conditions the SDD on them,

removes variables absent from the residual support, and identifies conditions that must occur in every remaining reason. The unresolved literals are then enumerated using a counterexample-guided minimal-hitting-set search, adapting the implicit hitting-set approach used for formal explanations (Ignatiev et al., 2019). Failed candidates produce conflicts that restrict subsequent search, while successful candidates yield prime implicants contained in μ. By Proposition 3, repeating this process over all nearest instances returns exactly the prime implicants contained in the nearest instances, and so exactly the reasons in $\min(PI(\Delta), \leq_{\omega}^{dis})$.

Once a reason τ is returned, it is weakened to $\tau \setminus dis_{\tau}(\omega)$ and, if selected, incorporated into Δ using the polynomial-time SDD Apply operation (Darwiche, 2011). Correctness of the procedure is proved in Appendix B.2, and a worked example is given in Appendix C.

The minimum-distance pass takes $O(|D|)$ time, but complete tie recovery remains exponential in the worst case, with up to 2^{r_μ} residual candidates for a nearest model μ, where r_μ is the number of optional literals remaining after preprocessing and n is the number of features. The bound $r_\mu \leq n - d^\star$ always holds; conditioning, support reduction, and necessary-condition detection can reduce the residual search space further. We evaluate empirically the performance of this procedure against naive enumeration of prime implicants in the following section.

5 Experimental Evaluation

We evaluate the procedure on random 3-CNF classifiers, a standard and controllable family of Boolean formulas whose difficulty can be varied through the clause-to-variable ratio. We use $n \in \{3, \ldots, 8\}$ and different density values $\rho \in \{2.0, 3.0, 3.8, 4.0, 4.2, 4.26, 4.4, 4.6, 5.0, 6.0\}$, covering sparse, near-transition, and denser regimes. In random 3-SAT, the satisfiability transition occurs around density 4.26 (Crawford & Auton, 1996). For each (n, ρ) combination we generate 25 classifiers, providing repeated instances at each setting while keeping exhaustive comparison feasible. Each classifier is compiled into an SDD using the same right-linear vtree, so differences in runtime are attributable to the enumeration procedure rather than to vtree selection. We evaluate all negatively classified assignments, since these are precisely the inputs for which a revision is requested, constituting 120,272 classifier-instance pairs. The proposed method is compared with an exhaustive baseline that enumerates $PI(\Delta)$ before selecting the minimum-disagreement reasons; this both provides an exact reference for correctness and directly measures the cost avoided by restricting enumeration to the relevant reasons.

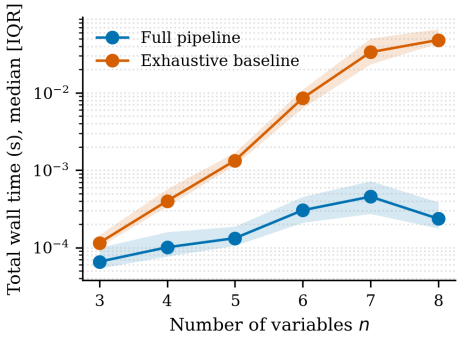


Figure 2: Median total time for complete recovery of the minimum-disagreement reasons. Shaded regions show the interquartile range.

Figure 2 shows the total time required to recover all minimum-disagreement reasons as the number of variables increases. The exhaustive baseline grows rapidly, while the full pipeline remains in the sub-millisecond range at the median across the tested values of n. The separation becomes increasingly pronounced as the instances grow, indicating that avoiding the materialisation of the complete set $PI(\Delta)$ can substantially reduce the work needed to recover only the minimum-disagreement reasons. These results are consistent with the intended advantage of restricting enumeration to nearest models and their residual reason spaces. However, the current experiment covers only $n = 3$-8;

larger variable ranges are needed to determine whether this behaviour persists and to draw stronger conclusions about scalability.

6 Future Work

An important next step is to connect the symbolic editing framework more closely to learned classifiers. When the classifier itself is symbolic, such as a decision tree, an edit can be applied directly after compilation. For other learned models, a Boolean representation may instead be compiled or extracted for explanation Shih et al. (2018); Huang et al. (2022); Darwiche & Hirth (2023). Our guarantees then concern this symbolic representation; translating an edit back into a change of the original model is model-dependent and remains an open problem.

The framework could also be extended beyond Boolean unary edits. Relevant directions include multi-valued and continuous features, multi-class classification, batch or iterated edits, and richer forms of explanation. Non-Boolean features may be encoded symbolically, but a principled account is needed for translating edited reasons back into meaningful constraints over the original feature space. Finally, the interactive interpretation should be evaluated with users through an interface that presents tied minimal corrections and elicits which generalisation is intended. On the computational side, experiments over larger variable ranges and more diverse classifier families would help determine when the residual enumeration becomes the dominant bottleneck.

7 Conclusion

We started from the insight that if a classifier's decisions can be explained in human-understandable terms, then these explanations can also be corrected. Moreover, these corrections can guide how the classifier itself should be modified. This led to our research question: How can a classifier be modified to incorporate user-provided corrections to its explanations?

We addressed this question by reinterpreting the edit operators introduced by Schwind et al. (2023). We introduced classifier revision as a process of correcting the sufficient reasons underlying a decision. This makes the effect of an edit explicit: rather than merely requiring that ω change classification, the user determines which explanation should be changed and therefore how the correction generalises to other instances. Our two representation theorems establish exact correspondences between constraints on reason adjustment and the postulates for positive unary edits and minimal change, respectively. Within the latter class, our minimal-weakening construction formalises one conservative way of editing the classifier, by modifying only minimum-disagreement reasons and removing only the literals that conflict with ω. Minimal weakening should not be interpreted as the only meaningful notion of reason-level change. Different applications may prefer other criteria for selecting or adjusting reasons.

For SDD-compiled classifiers, we further showed how the minimum-disagreement sufficient reasons can be recovered without first enumerating all prime implicants. The procedure restricts attention to nearest accepted instances and reduces the remaining search before performing counterexample-guided enumeration. Although complete tie recovery remains exponential in the worst case, the experiments show a substantial advantage over exhaustive prime-implicant enumeration on the tested instances. Together, these results provide a formal and computational basis for using explanations not only to justify classifier decisions, but also to guide their correction.

AI Use Statement

We used generative AI tools for proofreading, assistance with checking and revising mathematical proofs, and generating code for the experimental setup. The authors take responsibility for the final content of the paper and accompanying code. The authors reviewed the AI-assisted text and proofs and verified the generated code.

Reproducibility Statement

Source code for the experiments is provided in an anonymised ZIP archive submitted as supplementary material. The experimental setup is described in Section 5, and proofs and algorithm details are provided in Appendices A and B.

References

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985. doi: 10.2307/2274239. URL https://doi.org/10.2307/2274239.

James M. Crawford and Larry D. Auton. Experimental results on the crossover point in random 3-sat. *Artif. Intell.*, 81(1-2):31–57, 1996. doi: 10.1016/0004-3702(95)00046-1. URL https://doi.org/10.1016/0004-3702(95)00046-1.

Adnan Darwiche. On the tractable counting of theory models and its application to truth maintenance and belief revision. *J. Appl. Non Class. Logics*, 11(1-2):11–34, 2001. doi: 10.3166/JANCL.11.11-34. URL https://doi.org/10.3166/jancl.11.11-34.

Adnan Darwiche. SDD: A new canonical representation of propositional knowledge bases. In Toby Walsh (ed.), *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pp. 819–826. IJCAI/AAAI, 2011. doi: 10.5591/978-1-57735-516-8/IJCAI11-143. URL https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-143.

Adnan Darwiche. Logic for explainable AI. *CoRR*, abs/2305.05172, 2023. doi: 10.48550/ARXIV.2305.05172. URL https://doi.org/10.48550/arXiv.2305.05172.

Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang (eds.), *ECAI 2020*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 712–720. IOS Press, 2020. doi: 10.3233/FAIA200158. URL https://doi.org/10.3233/FAIA200158.

Adnan Darwiche and Auguste Hirth. On the (complete) reasons behind decisions. *J. Log. Lang. Inf.*, 32(1):63–88, 2023. doi: 10.1007/s10849-022-09377-8. URL https://doi.org/10.1007/s10849-022-09377-8.

Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *J. Artif. Intell. Res.*, 17:229–264, 2002. doi: 10.1613/JAIR.989. URL https://doi.org/10.1613/jair.989.

Peter Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, 1988.

Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin C. Cooper, Nicholas Asher, and João Marques-Silva. Tractable explanations for d-dnnf classifiers. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 5719–5728. AAAI Press, 2022. doi: 10.1609/AAAI.V36I5.20514. URL https://doi.org/10.1609/aaai.v36i5.20514.

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

Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artif. Intell.*, 52(3):263–294, 1992. doi: 10.1016/0004-3702(91)90069-V. URL https://doi.org/10.1016/0004-3702(91)90069-V.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

João Marques-Silva. Logic-based explainability: Past, present & future. *CoRR*, abs/2406.11873, 2024. doi: 10.48550/ARXIV.2406.11873. URL https://doi.org/10.48550/arXiv.2406.11873.

Antonio Rago and Maria Vanina Martinez. Advancing interactive explainable ai via belief change theory. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*, KR '24, 2024. ISBN 978-1-956792-05-8. doi: 10.24963/kr.2024/87. URL https://doi.org/10.24963/kr.2024/87.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Nicolas Schwind, Katsumi Inoue, and Pierre Marquis. Editing boolean classifiers: A belief change perspective. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 6516–6524. AAAI Press, 2023. doi: 10.1609/AAAI.V37I5.25801. URL https://doi.org/10.1609/aaai.v37i5.25801.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. In Jérôme Lang (ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 5103–5111. ijcai.org, 2018. doi: 10.24963/IJCAI.2018/708. URL https://doi.org/10.24963/ijcai.2018/708.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Comput. Surv.*, 56(12):312:1–312:42, 2024. doi: 10.1145/3677119. URL https://doi.org/10.1145/3677119.

A Proofs of the Main Representation and Characterisation Results

This appendix proves the representation theorems and the minimal weakening result from Section 3, followed by the nearest-model characterisation used in Section 4. Appendix B gives the enumeration algorithms and their correctness proofs; Appendix C illustrates the computation on a concrete classifier.

A.1 Representation and Minimal Weakening

Proof of Theorem 1. Suppose C satisfies (C1) and (C2). Condition (C1) gives $\omega \models \Delta \oplus_C \omega$, establishing (P1). If $\omega \models \Delta$, some member of $PI(\Delta)$ applies to ω, so (C2) leaves that set unchanged and gives $\Delta \oplus_C \omega \equiv \Delta$, establishing (P2). Equivalent classifiers have identical prime-implicant sets, which establishes (P3). These properties also hold for any operator represented by C. Conversely, suppose $\oplus$ satisfies (P1)–(P3). For any set S, put $\phi_S = \bigvee S$ and define

$$C(S, \omega) = \begin{cases} S, & \omega \models \phi_S, \\ PI(\phi_S \oplus \omega), & \omega \not\models \phi_S. \end{cases}$$

In the first case, some member of S applies to ω. In the second, (P1) ensures that ω satisfies $\phi_S \oplus \omega$ and hence one of its prime implicants. Thus (C1) holds, and (C2) holds by construction. By (P2) in the first case and the prime-implicant representation in the second, $\bigvee C(S, \omega) \equiv \phi_S \oplus \omega$. For $S = PI(\Delta)$, we have $\phi_S \equiv \Delta$, so (P3) gives $\bigvee C(PI(\Delta), \omega) \equiv \Delta \oplus \omega$. Hence C represents $\oplus$. □

Proof of Theorem 2. Suppose a minimal adjustment function C represents $\oplus$. Theorem 1 gives (P1)–(P3). For (PW), every model of Δ satisfies some $\tau \in PI(\Delta)$. By (CW), an output term $\rho \subseteq \tau$ exists, so this model also satisfies $\Delta \oplus \omega$.

For (PR), suppose $\tau' \models \Delta \oplus \omega$ but $\tau' \not\models \Delta$. Some $\sigma \in C(PI(\Delta), \omega)$ applies to τ'. This σ cannot belong to $PI(\Delta)$, since it would then imply Δ. Thus (CR) gives $\sigma \subseteq \omega$. We also have $\sigma \subseteq \tau'$ and $\sigma \models \Delta \oplus \omega$, since C represents $\oplus$. Hence σ witnesses (PR).

For (PC), suppose $\Delta \models \eta$ and $\eta \subseteq \omega$. Every term in $PI(\Delta)$ contains η. An output term retained from $PI(\Delta)$ therefore contains η, and any other output term contains η by (CC). Hence $\Delta \oplus \omega \models \eta$. This argument also applies when $PI(\Delta) = \emptyset$.

Conversely, suppose $\oplus$ is a minimal change operator. For any set S of terms, put $\phi_S = \bigvee S$ and $F_{S,\omega} = \phi_S \oplus \omega$. Let

$$R_\omega(F) = \{\rho \in PI(F) : \rho \subseteq \omega\}$$

be the prime implicants of F that apply to the target, and define

$$C(S, \omega) = \begin{cases} S, & \omega \models \phi_S, \\ S \cup R_\omega(F_{S,\omega}), & \omega \not\models \phi_S. \end{cases} \tag{5}$$

We first show that $\bigvee C(S, \omega) \equiv F_{S,\omega}$. In the first case this follows from (P2). In the second case, (PW) gives $\phi_S \models F_{S,\omega}$, and every member of $R_\omega(F_{S,\omega})$ implies $F_{S,\omega}$. Conversely, a model of $F_{S,\omega}$ either satisfies ϕ_S, or is newly accepted. In the latter case, (PR) provides an implicant of $F_{S,\omega}$ applying to both this model and ω. Shrinking that implicant yields a member of $R_\omega(F_{S,\omega})$ covering the model. Thus

$$F_{S,\omega} \equiv \phi_S \vee \bigvee R_\omega(F_{S,\omega}) \equiv \bigvee C(S, \omega).$$

For $S = PI(\Delta)$, syntax independence (P3) gives $F_{S,\omega} \equiv \Delta \oplus \omega$, so C represents $\oplus$.

We check the constraints on every input S. In the first branch, some member of S applies to ω; in the second, (P1) gives $\omega \models F_{S,\omega}$, so a prime implicant of $F_{S,\omega}$ applies to ω and belongs to $R_\omega(F_{S,\omega})$. Thus (C1) holds. Condition (C2) follows from the first branch. Every original term is retained, giving (CW), and every added term is a subset of ω, giving (CR).

Finally, suppose $\rho \in C(S, \omega) \setminus S$, $\eta \subseteq \omega$, and $\eta \subseteq \tau$ for every $\tau \in S$. Then $\phi_S \models \eta$, including when $S = \emptyset$. By (PC), $F_{S,\omega} \models \eta$. Since $\rho \in R_\omega(F_{S,\omega})$, we have $\rho \models F_{S,\omega} \models \eta$. Entailment between consistent terms gives $\eta \subseteq \rho$, proving (CC). □

Proof of Proposition 2. If $S = \emptyset$, the output $\{\omega\}$ satisfies all five constraints: (C2) and (CW) have false antecedents, and every term $\eta \subseteq \omega$ is contained in the only output term, as required by (CC).

Suppose $S \neq \emptyset$. Each selected term τ is replaced by $\tau \setminus dis_\tau(\omega) = \tau \cap \omega$. Since S_w is nonempty, some output term applies to ω, giving (C1). If an input term already applies, the minimum disagreement is zero, so every selected term is unchanged and $C(S, \omega) = S$, giving (C2). Every input term is retained or weakened, giving (CW), and every newly introduced term is a subset of ω, giving (CR). For (CC), any $\eta \subseteq \omega$ contained in every input term is contained in $\tau \cap \omega$ for each selected τ. Thus every newly introduced term retains η. The operator properties follow from Theorem 2. □

A.2 Nearest-Model Characterisation

The next two lemmas connect literal disagreement with Hamming distance. Together, they establish Proposition 3: the candidate reasons for minimal weakening can be recovered from the nearest accepted instances. These lemmas are also used in the algorithm proofs in Appendix B.2.

Lemma 1 (A reason induces a model at its disagreement distance)**.** *Let $\tau \in PI(\Delta)$ and let $|dis_\tau(\omega)| = k$. Construct μ by assigning the literals occurring in τ as specified by τ, and copying ω on all remaining features. Then $\mu \models \Delta$ and $\mathrm{Ham}(\mu, \omega) = k$.*

Proof. Since $\mu \models \tau$ and $\tau \models \Delta$, we have $\mu \models \Delta$. The only features on which μ differs from ω are those represented by literals in $dis_\tau(\omega)$, giving $\mathrm{Ham}(\mu, \omega) = k$. □

Lemma 2 (Minimum model distance equals minimum reason disagreement)**.** *Let Δ be satisfiable, let ω be an instance, and let $d^\star = \min_{\mu \models \Delta} \mathrm{Ham}(\mu, \omega)$. Then $d^\star = \min_{\tau \in PI(\Delta)} |dis_\tau(\omega)|$. Thus, for every $\tau \in PI(\Delta)$, $\tau \in \min(PI(\Delta), \leq_\omega^{dis})$ $\iff$ $|dis_\tau(\omega)| = d^\star$.*

Proof. By Lemma 1, every prime implicant τ induces an accepted model at distance $|dis_\tau(\omega)|$, so $d^\star \leq \min_{\tau \in PI(\Delta)} |dis_\tau(\omega)|$. Conversely, any model $\mu \models \Delta$ satisfies some prime implicant $\tau \subseteq \mu$. Therefore $|dis_\tau(\omega)| \leq \mathrm{Ham}(\mu, \omega)$. Applying this to a nearest model gives the reverse inequality. The second claim follows directly. □

Proof of Proposition 3. Suppose $\tau \in \min(PI(\Delta), \leq^{dis}_\omega)$. By Lemma 2, $|dis_\tau(\omega)| = d^\star$. Lemma 1 then constructs a model $\mu \models \Delta$ containing τ with $\mathrm{Ham}(\mu, \omega) = d^\star$. For the other direction, suppose $\tau \subseteq \mu$ for some $\mu \models \Delta$ with $\mathrm{Ham}(\mu, \omega) = d^\star$. Since every disagreement literal of τ must also be a feature on which μ differs from ω, $|dis_\tau(\omega)| \leq d^\star$. By Lemma 2, no prime implicant can have disagreement smaller than $d^\star$. So $|dis_\tau(\omega)| = d^\star$, and therefore $\tau \in \min(PI(\Delta), \leq^{dis}_\omega)$. □

B Enumeration of Minimum-Disagreement Reasons

This appendix expands the procedure outlined in Section 4. We first give the two algorithms and explain their preprocessing and search steps, then prove that they return exactly the required reasons without duplicates. Appendix C follows the same steps on a small classifier.

B.1 Algorithms and Search Procedure

Throughout this appendix, D is an SDD representing a satisfiable classifier Δ, ω is a rejected instance, and $d^\star = \min_{\mu \models \Delta} \mathrm{Ham}(\mu, \omega)$. Algorithm 1 first finds $d^\star$, enumerates the accepted instances at that distance, and recovers all prime implicants contained in each such instance. It returns the original reasons for selection by the user; weakening the selected reasons is the subsequent adjustment step from Definition 8.

Algorithm 1 EnumerateReasons(Δ, ω)

Require: satisfiable compiled classifier Δ, rejected instance $\omega \not\models \Delta$
Ensure: all reasons in $\min(PI(\Delta), \leq^{dis}_\omega)$, each once

```
d* ← MinDistance(D, ω)
for all μ ∈ ModelsAtMinDistance(D, ω, d*) do
    for all τ ∈ ReasonsFromModel(μ, D, ω) do
        yield τ
    end for
end for
```

Minimum distance and nearest models. For the fixed target ω, let $d(\alpha) = \textsc{MinDistance}(\alpha)$ denote the minimum Hamming distance to a model of the SDD node α, counting only variables represented below that node. Writing $\mathrm{dn}(\alpha)$ for its decision elements, we compute this distance bottom-up:

$$d(\alpha) = \begin{cases} 0, & \alpha = \top \text{ or } \alpha = l \text{ with } \omega \models l, \\ 1, & \alpha = l \text{ with } \omega \not\models l, \\ \infty, & \alpha = \bot, \\ \min\limits_{(p,s) \in \mathrm{dn}(\alpha)} \big(d(p) + d(s)\big), & \text{otherwise.} \end{cases}$$

Each SDD node is processed once, giving $O(|D|)$ time. Starting from the root value $d^\star$, ModelsAtMinDistance follows only decision elements whose local costs attain the stored minimum and therefore reconstructs exactly the nearest accepted instances, each once. Unconstrained features retain their value in ω.

Reasons contained in a nearest model. A nearest model μ may contain more than one minimum-disagreement prime implicant, so a single greedy shrink is insufficient. Algorithm 2 first removes choices whose role is already determined, then enumerates the remaining possibilities using counterexample-guided minimal hitting sets. Here $D \mid F_\mu$ denotes conditioning on the literals in F_μ, $\mathrm{Vars}(D_\mu)$ is the variable set of the residual, and $\mathrm{var}(l)$ is the variable of literal l.

The family $\mathcal{E}$ stores learned conflicts, and $\mathcal{B}$ stores successful optional sets whose supersets are blocked. A hitting set intersects every member of $\mathcal{E}$ and is minimal if no proper subset does so. The helper NEXTMINIMALHITTINGSET$(\mathcal{E}, \mathcal{B})$ returns such a minimal set $A \subseteq U_\mu$ containing no member of $\mathcal{B}$ as a subset, or $\perp$ if none exists. When $\mathcal{E}$ is empty, its unique minimal hitting set is $\emptyset$.

Algorithm 2 REASONSFROMMODEL(μ, D, ω)

Require: nearest model $\mu \models D$ with $\mathrm{Ham}(\mu, \omega) = d^\star$
Ensure: all $\tau \in PI(D)$ such that $\tau \subseteq \mu$

$F_\mu \leftarrow \{l \in \mu : \omega \not\models l\}$ {fixed disagreement literals}
$D_\mu \leftarrow D \mid F_\mu$ {conditioned residual}
$O_\mu \leftarrow \{l \in \mu \setminus F_\mu : \mathrm{var}(l) \in \mathrm{Vars}(D_\mu)\}$
$N_\mu \leftarrow \{l \in O_\mu : O_\mu \setminus \{l\} \not\models D_\mu\}$ {necessary conditions}
$U_\mu \leftarrow O_\mu \setminus N_\mu$ {optional literals}
$\mathcal{E} \leftarrow \emptyset, \quad \mathcal{B} \leftarrow \emptyset$
loop
  $A \leftarrow$ NEXTMINIMALHITTINGSET$(\mathcal{E}, \mathcal{B})$
  if $A = \perp$ **then**
    stop
  else if $(N_\mu \cup A) \models D_\mu$ **then**
    yield $\tau \leftarrow F_\mu \cup N_\mu \cup A$ {return the completed prime implicant}
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{A\}$ {block A and its supersets}
  else
    choose $\nu \models (N_\mu \cup A) \wedge \neg D_\mu$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{\{l \in U_\mu : \nu \not\models l\}\}$ {add learned conflict}
  end if
end loop

The algorithm starts from $F_\mu = \{l \in \mu : \omega \not\models l\}$, which contains the literals on which μ differs from ω. Because μ is nearest, every prime implicant contained in μ must contain these literals (Lemma 4), so they are fixed immediately. Conditioning on F_μ yields D_μ; variables absent from its support cannot be required by a reason extending F_μ (Lemma 5). Finally, a remaining literal l is a *necessary condition* if $O_\mu \setminus \{l\} \not\models D_\mu$. Such literals form N_μ and occur in every remaining reason (Lemma 6). The search therefore only ranges over

$$U_\mu = O_\mu \setminus N_\mu, \qquad r_\mu = |U_\mu| \leq n - d^\star.$$

Every remaining reason has the form

$$\tau = F_\mu \cup N_\mu \cup A, \qquad A \subseteq U_\mu.$$

Rather than testing all subsets A, the procedure learns conflicts from failed candidates. If $N_\mu \cup A$ does not entail D_μ, a countermodel ν yields $E_\nu = \{l \in U_\mu : \nu \not\models l\}$. Every valid reason must contain at least one literal from E_ν; otherwise the same countermodel would falsify the residual while satisfying the reason. Subsequent candidates are therefore minimal hitting sets of the learned conflicts. Successful candidates are returned and blocked together with their supersets, since a strict superset cannot define a new prime implicant. The loop terminates when no unblocked minimal hitting set remains.

B.2 CORRECTNESS AND COMPLETENESS OF THE ENUMERATION OF REASONS

The proof follows the two levels of the procedure. First, we show that each minimum-disagreement reason belongs to exactly one nearest model, which will exclude duplicates across models. We then justify the preprocessing, prove that Algorithm 2 returns all and only the prime implicants contained in a given nearest model, and combine these results to establish correctness of Algorithm 1.

Lemma 3 (A minimum-disagreement reason determines one nearest model)**.** *Let $\tau \in PI(\Delta)$ satisfy $|dis_\tau(\omega)| = d^\star$. There is exactly one model μ at distance $d^\star$ from ω containing τ: the model constructed in Lemma 1.*

Proof. Any model containing τ must already differ from ω on the $d^\star$ features in $dis_\tau(\omega)$. Any additional difference would increase its distance beyond $d^\star$. All remaining features must therefore retain their value in ω, uniquely determining μ. □

B.2.1 Correctness of Preprocessing

The following lemmas justify fixing the disagreement literals, discarding variables absent from the residual, and retaining necessary conditions. Each step preserves every prime implicant contained in the chosen nearest model.

Lemma 4 (Flipped literals occur in every contained reason)**.** *Let $\mu \models \Delta$ be a nearest model and $F_\mu = \{l \in \mu : \omega \not\models l\}$. Then $F_\mu \subseteq \tau$ for every $\tau \in PI(\Delta)$ with $\tau \subseteq \mu$.*

Proof. If such a τ omitted some $l \in F_\mu$, then $|dis_\tau(\omega)| < |F_\mu| = d^\star$, contradicting Lemma 2. □

Lemma 5 (Variables outside the residual support are unnecessary)**.** *Let $D_\mu = D \mid F_\mu$. If $x \notin \mathrm{Vars}(D_\mu)$, then no prime implicant $\tau \subseteq \mu$ contains a literal on x outside F_μ.*

Proof. Every such τ contains F_μ by Lemma 4, so $\tau \setminus F_\mu \models D_\mu$. Since D_μ does not depend on x, removing a literal on x preserves this entailment, contradicting the primality of τ. □

Lemma 6 (Necessary conditions occur in every contained reason)**.** *Let $O_\mu = \{l \in \mu \setminus F_\mu : \mathrm{var}(l) \in \mathrm{Vars}(D_\mu)\}$. If $l \in O_\mu$ satisfies $O_\mu \setminus \{l\} \not\models D_\mu$, then every $\tau \in PI(\Delta)$ with $\tau \subseteq \mu$ contains l.*

Proof. Suppose such a τ omitted l. By Lemmas 4 and 5, $\tau \setminus F_\mu \subseteq O_\mu \setminus \{l\}$. Since $\tau \setminus F_\mu \models D_\mu$, the stronger term $O_\mu \setminus \{l\}$ must also entail D_μ, a contradiction. □

Let N_μ denote the set of these necessary conditions and define $U_\mu = O_\mu \setminus N_\mu$. Every prime implicant contained in μ therefore has the form $\quad F_\mu \cup N_\mu \cup A, \quad A \subseteq U_\mu$.

B.2.2 Correctness of ReasonsFromModel

It remains to show that the search over optional literals neither introduces non-prime reasons nor misses valid ones. Learned conflicts enforce necessity, while blocking returned candidates prevents repetition.

Proposition 4. *Assume that* NextMinimalHittingSet *completely enumerates the currently unblocked minimal hitting sets of the accumulated conflict family. Then* ReasonsFromModel(μ, D, ω) *yields exactly $\{\tau \in PI(\Delta) : \tau \subseteq \mu\}$, each once.*

Proof. We prove soundness and completeness.

Soundness. Suppose the procedure yields $\tau = F_\mu \cup N_\mu \cup A$ after verifying $N_\mu \cup A \models D_\mu$. Hence $\tau \models \Delta$ and $\tau \subseteq \mu$.

Every implicant contained in μ contains a prime implicant contained in μ. Thus no literal in F_μ or N_μ can be removed from τ, by Lemmas 4 and 6. Now consider $l \in A$. Because A is a minimal hitting set, some learned conflict E is hit by A only through l. Let ν_E be the countermodel that produced E. Then $\nu_E \models N_\mu$, $\nu_E \not\models D_\mu$, and ν_E satisfies every literal in $A \setminus \{l\}$. So $N_\mu \cup (A \setminus \{l\}) \not\models D_\mu$. Removing any literal therefore destroys implicanthood, so τ is prime.

Completeness. Let $\tau \in PI(\Delta)$ with $\tau \subseteq \mu$, and write $A_\tau = \tau \setminus (F_\mu \cup N_\mu)$. Every learned conflict must intersect A_τ; otherwise the countermodel that produced that conflict would satisfy $N_\mu \cup A_\tau$ while falsifying D_μ, contradicting $\tau \models \Delta$.

If A_τ is not yet a minimal hitting set, some strict subset $H \subset A_\tau$ is. Such an H cannot be blocked by an already returned reason, since that would give an implicant strictly contained in the prime implicant τ. If H is tested, it cannot succeed for the same reason; it therefore produces a new conflict that excludes H. If another conflict excludes it first, the same progress has already occurred. Because A_τ has only finitely many strict subsets, eventually none remains a minimal hitting set. At that point A_τ itself is an unblocked minimal hitting set and is tested and returned. Lastly, once an

optional set A is returned, it and all its supersets are blocked. Therefore the same prime implicant cannot be returned twice.

□

B.2.3 Correctness of the Overall Enumeration

We now combine the correctness of the per-model search with the nearest-model characterisation. Soundness ensures that every returned reason has minimum disagreement; completeness and the unique-nearest-model property ensure that every such reason is returned exactly once.

Proposition 5 (Soundness)**.** *Every term τ returned by Algorithm 1 is a prime implicant of Δ with $|dis_\tau(\omega)| = d^\star$.*

Proof. Every returned term originates from a nearest model μ and a reason τ returned by ReasonsFromModel(μ, D, ω). By Proposition 4, $\tau \in PI(\Delta)$ and $\tau \subseteq \mu$. Therefore $|dis_\tau(\omega)| \leq \mathrm{Ham}(\mu, \omega) = d^\star$. Lemma 2 rules out a smaller value, so equality holds. Algorithm 1 returns this reason τ unchanged. □

Proposition 6 (Minimum-level completeness)**.** *Algorithm 1 returns exactly $\min(PI(\Delta), \leq_\omega^{dis})$, with each reason returned once.*

Proof. Soundness follows from Proposition 5 and Lemma 2.

Let $\tau \in \min(PI(\Delta), \leq_\omega^{dis})$. Then $|dis_\tau(\omega)| = d^\star$. By Lemma 1, τ induces a nearest model μ, which is enumerated by ModelsAtMinDistance. Proposition 4 then guarantees that ReasonsFromModel(μ, D, ω) returns τ. Lemma 3 shows that no other nearest model can contain the same τ, while Proposition 4 prevents duplicates within μ. Therefore every minimum-disagreement reason is generated exactly once. □

C Worked Example of the Enumeration of Reasons

This example traces both algorithms in Appendix B: nearest-model enumeration, preprocessing for each model, and the search over optional literals. The final step shows how a returned reason becomes an edit after the user selects it.

We illustrate Algorithm 1 on $\Delta = (a \wedge b) \vee (c \wedge d) \vee (\overline{a} \wedge \overline{c})$, over features $\{a, b, c, d\}$. Its prime implicants are $PI(\Delta) = \{cd, bd, b\overline{c}, ab, \overline{a}d, \overline{ac}\}$. Consider the negatively classified instance $\omega = \overline{abcd}$. The minimum-disagreement reasons we need to obtain are $\{cd, \overline{a}d, \overline{ac}\}$, each disagreeing with ω on one literal; the remaining prime implicants have disagreement 2.

Minimum distance and nearest models. The first stage of Algorithm 1 computes $d^\star = 1$ and enumerates the accepted instances at this distance:

$$\mu_1 = \overline{abcd}, \qquad \mu_2 = \overline{a}bcd.$$

The procedure then calls ReasonsFromModel for each of them.

Reducing the search for each nearest model. For μ_2, the only literal flipped relative to ω is d, so $F_{\mu_2} = \{d\}$. Conditioning on it gives $D_{\mu_2} \equiv (a \wedge b) \vee c \vee (\overline{a} \wedge \overline{c})$. The remaining model literals over its support are $O_{\mu_2} = \{\overline{a}, \overline{b}, c\}$. None is necessary to every reason: removing any one of them still leaves a set that entails D_{μ_2}. Hence $N_{\mu_2} = \emptyset$ and $U_{\mu_2} = \{\overline{a}, \overline{b}, c\}$.

For μ_1, the flipped literal is $\overline{c}$, so $F_{\mu_1} = \{\overline{c}\}$. Conditioning yields $D_{\mu_1} \equiv (a \wedge b) \vee \overline{a}$ and $O_{\mu_1} = \{\overline{a}, \overline{b}\}$. Here $\overline{a}$ is necessary: $\{\overline{b}\}$ does not entail the residual, whereas $\{\overline{a}\}$ does. Thus $N_{\mu_1} = \{\overline{a}\}$ and $U_{\mu_1} = \{\overline{b}\}$.

Counterexample-guided enumeration. For μ_2, the hitting-set search begins with $A = \emptyset$, which does not entail D_{μ_2}. A countermodel is $\nu = (a, b, c) = (1, 0, 0)$. It falsifies the optional literals $\overline{a}$ and c, producing the conflict $E_\nu = \{\overline{a}, c\}$. Its minimal hitting sets are therefore $\{\overline{a}\}$ and $\{c\}$, and both entail the residual. Reintroducing the fixed literal d yields the two reasons $\overline{a}d$ and cd.

For μ_1, the empty optional candidate already succeeds because $\overline{a} \models D_{\mu_1}$. Combining it with the fixed literal $\overline{c}$ therefore yields the reason $\overline{ac}$.

Result. Across the two nearest models, Algorithm 1 returns exactly the reasons $\{cd, \overline{a}d, \overline{ac}\} = \min(PI(\Delta), \leq_{\omega}^{dis})$. If, for example, cd is selected, weakening removes its disagreement literal d, giving $cd \setminus dis_{cd}(\omega) = c$ and the revised classifier $\Delta \vee c$.

The example illustrates why both parts of the enumeration are required: different nearest models can contain different minimum-disagreement reasons, while a single nearest model can contain several such reasons. The counterexample-guided hitting-set stage recovers all of them without enumerating the higher-disagreement prime implicants.