
Model Reconciliation via Cost-Optimal Explanations in Probabilistic Logic Programming

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 In human-AI interaction, effective communication relies on aligning the AI agent’s
2 model with the human user’s mental model – a process known as model reconcilia-
3 tion. However, existing model reconciliation approaches predominantly assume
4 deterministic models, overlooking the fact that human knowledge is often uncertain
5 or probabilistic. To bridge this gap, we present a probabilistic model reconciliation
6 framework that resolves inconsistencies in MPE outcome probabilities between
7 an agent’s and a user’s models. Our approach is built on probabilistic logic pro-
8 gramming (PLP) using ProbLog, where explanations are generated as cost-optimal
9 model updates that reconcile these probabilistic differences. We develop two search
10 algorithms – a generic and an optimized version, with the latter guided by theo-
11 retical insights for scalability. Our approach is validated through a user study on
12 how explanation types impact user understanding and computational experiments
13 showing the optimized search consistently outperforms the generic.

14 1 Introduction

15 In human-AI interaction, effective communication relies on aligning the AI agent’s model with the
16 human user’s mental model, as mismatched understandings can make the agent’s behavior seem
17 inexplicable [1]. Model reconciliation offers a powerful explainable AI (XAI) approach by adjusting
18 the human’s model to align with the agent’s understanding [2]. For instance, in planning, it explains
19 why an agent’s actions are valid in its model but not in the human’s [2]. However, existing model
20 reconciliation methods assume deterministic user beliefs, treating them as fixed or drawn from a
21 set of distinct models [3, 4]. This overlooks the uncertainty and graded beliefs typical of human
22 knowledge. In reality, humans maintain probabilistic beliefs – degrees of confidence rather than
23 absolute truths – and ignoring this uncertainty can lead to unconvincing or misaligned explanations.

24 Probabilistic reasoning addresses uncertainty, capturing graded beliefs and uncertain outcomes.
25 Inference methods like most probable explanation (MPE) and maximum a posteriori (MAP) are
26 key: MPE finds the most likely scenario given evidence, while MAP identifies the most probable
27 hypothesis [5]. These methods are widely applied, from Bayesian networks to probabilistic logic
28 models. However, model reconciliation has yet to fully account for probabilistic beliefs. An
29 agent may base decisions on an MPE outcome, while a human might disagree due to different
30 probabilistic assumptions. Existing approaches treat user knowledge as a set of deterministic models
31 with probabilities [6], rather than a unified probabilistic model. Bridging this gap – reconciling
32 probabilistic model differences between an agent and a human – remains an open challenge.

33 Despite this gap, model reconciliation has been extensively explored in logic-based systems. In
34 classical planning and answer set programming (ASP), explanations are generated by modifying
35 logical rules to align beliefs [3]. Meanwhile, probabilistic logic programming (PLP) frameworks like
36 ProbLog [7] offer a powerful way to handle uncertainty, combining logical rules with probabilistic
37 semantics. PLP supports diverse inference tasks, including MPE, MAP, and marginal probability

computation, making it a versatile tool for uncertain reasoning. However, while PLP excels at probabilistic reasoning, it has not been integrated with model reconciliation to generate explanations that reconcile probabilistic beliefs, leaving a critical gap in existing approaches.

In this paper, we introduce the first probabilistic model reconciliation framework within a PLP setting, leveraging ProbLog for its expressive power. Our approach allows an agent to reconcile differences between its probabilistic model and a human’s model by generating cost-optimal explanations that resolve inconsistencies in MPE outcome probabilities. Our key contributions are as follows:

- **Probabilistic Model Reconciliation:** We define model reconciliation under uncertainty, addressing inconsistencies in MPE outcome probabilities between an agent’s and a human’s ProbLog models.
- **Cost-Optimal Explanations:** We introduce a cost-based model where explanations are minimal updates that resolve probabilistic differences.
- **Algorithms:** We design two search algorithms for generating cost-optimal explanations – a generic search and an optimized search that prunes the search space using theoretical insights.
- **Comprehensive Evaluation:** We validate our approach with a user study on explanation costs and computational tests showing the optimized search’s superior performance.

2 Related Work

Model Reconciliation Problems (MRPs). MRPs have been widely studied in domains like automated planning, logic programming, and knowledge-base reasoning. In explainable planning [8], Chakraborti *et al.* defined it as aligning an agent’s (planning) model with a human’s by making minimal changes to the human model, using search methods like A* to balance explanation completeness and simplicity [2]. In logic programming, Son *et al.* reconciled answer-set programs by identifying minimal rule additions and deletions such that the program yield the same target conclusion [3]. For knowledge bases, Vasileiou *et al.* used a hitting-set approach to compute minimal sets of formulas that prove a target conclusion [4]. Beyond deterministic models, Sreedharan *et al.* addressed uncertainty about the human’s model [9] by focusing on scenarios where the human’s model is located within a space of possible human models that the agent has, and their method operates in that space to find explanations applicable to a set of possible models.

These methods share a common limitation: They assume user beliefs are deterministic or drawn from a fixed set of models. Importantly, uncertainty is handled as a collection of separate models with probabilities, not as a single model encoding probabilistic beliefs. As a result, these methods cannot reconcile differences when the agent’s and human’s models involve probability distributions over facts, rules, or outcomes. Our work addresses this gap by introducing a framework for probabilistic model reconciliation.

Probabilistic Logic Programming (PLP). PLP integrates probabilistic reasoning with the expressive power of logic programming, enabling the specification of complex probabilistic models. This line of research started with Poole [10], who introduced the first PLP framework by extending the logic programming language Prolog [11], and with Sato [12], whose distribution semantics became the basis for several PLP systems, such as PRISM [13], ICL [14], ProbLog [7], and LPAD [15]. The notion of explanation has been explored by the PLP community [16, 17], where explanations have been associated with possible worlds.¹ The most prominent task there is that of the most probable explanation (MPE), which consists of finding the world with the highest probability given some evidence [18]. However, a world does not show the chain of inferences of a given explanandum and it is not minimal by definition, since it usually includes a (possibly large) number of probabilistic facts whose truth value is irrelevant for the explanandum. An alternative approach is using the proof of an explanandum as an explanation [19], where a proof is a (minimal) partial world in which the query is true. In this case, one can easily ensure minimality, but even if the partial world contains no irrelevant facts, it is still not easy to determine the chain of inferences behind a given query. Finally, 20 have leveraged explanations in PLP as approximation techniques for more efficiently computing weighted model counting problems.

3 Background

We begin by reviewing the fundamental concepts of logic programming and its probabilistic extensions, with an emphasis on logical inference.

¹A possible world is a truth value assignment to the atoms in the language.

90 3.1 Logic Programming

91 An *atom* is an expression of the form $q(t_1, \dots, t_n)$, where q is a predicate of arity n , and each t_i is a
 92 *term*. A term t_i can be a *constant*, a *variable*, or a *functor* applied to other terms. A *literal* is either an
 93 atom or its negation $\neg q(t_1, \dots, t_n)$. An expression is said to be *ground* if it contains no variables.

94 Syntactically, a *normal clause program* – or *logic program* – is a set of *rules*. A *rule* r is an expression
 95 of the form $h :- b_1, \dots, b_n$, where h is an atom, referred to as the *head* of the rule, denoted by
 96 $\text{head}(r) = h$. The *body* of the rule consists of a conjunction of literals b_1, \dots, b_n , and is denoted
 97 by $\text{body}(r) = \{b_1, \dots, b_n\}$. The symbol ‘ $:-$ ’ represents logical implication (\leftarrow), and the comma ‘ $,$ ’
 98 denotes conjunction (\wedge). Thus, the rule states that h holds whenever all literals in the body are
 99 satisfied. If $n = 0$, meaning the rule has an empty body, the rule is called a *fact*.

100 3.2 Probabilistic Logic Programming

101 **Syntax.** A ProbLog program \mathcal{M} consists of a set of probabilistic facts \mathcal{F} and a set of logic rules
 102 \mathcal{R} . Formally, the set of probabilistic facts can be written as $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$, where each f_i
 103 is a ground fact. A *probabilistic fact*, written as $p_i :: f_i$, assigns a probability p_i to the fact f_i , i.e.,
 104 $P(f_i) = p_i$. Each fact is associated with a probability value.

105 Logic rules define deterministic dependencies between atoms (for simplicity, we assume that all
 106 atoms are ground). An atom that unifies with a probabilistic fact is called a probabilistic atom,
 107 whereas an atom that unifies with the head of a rule is referred to as a derived atom. We assume that
 108 the sets of probabilistic and derived atoms are disjoint.

109 **Semantics.** Each ground probabilistic fact $p_i :: f_i$ defines an *atomic choice*, in which f_i is either
 110 included (with probability p_i) or excluded (with probability $1 - p_i$). A *total choice* is formed by
 111 making an atomic choice for each fact in \mathcal{F} , resulting in a subset $\mathcal{C} \subseteq \mathcal{F}$ of the selected facts. If there
 112 are n probabilistic facts, the number of possible total choices is 2^n . From those choices, we derive
 113 the remaining atoms by applying the logic rules.

114 The probability of a total choice \mathcal{C} is computed by treating all atomic choices as independent events:

$$P(\mathcal{C}) = \prod_{f_i \in \mathcal{C}} p_i \cdot \prod_{f_i \in \mathcal{F} \setminus \mathcal{C}} (1 - p_i). \quad (1)$$

115 3.3 Inference

116 Given a ground atom q , referred to as a *query*, the *relevant ground program* $\mathcal{M}(q)$ denotes the
 117 minimal subset of the grounded version of the original program \mathcal{M} that is sufficient to derive q .
 118 Specifically, $\mathcal{M}(q)$ is constructed via backward reasoning, starting from q and recursively identifying
 119 all probabilistic facts and rules necessary for its derivation. This process ensures that only the
 120 components relevant to the query are retained, thereby preserving correctness while enhancing the
 121 efficiency of probabilistic inference.

122 In the context of model reconciliation, the *Most Probable Explanation* (MPE) inference is employed
 123 to identify the most probable set of assumptions that explain why a given query q holds. This supports
 124 alignment between an agent and a human user by providing interpretable explanations.

125 Formally, the MPE inference is defined as:

$$\text{MPE}(q \mid \mathcal{M}) = \arg \max_{\mathcal{C}(q) \subseteq \mathcal{F}(q)} P(\mathcal{C}(q) \mid q), \quad (2)$$

126 where $\mathcal{F}(q)$ is the set of ground probabilistic facts in $\mathcal{M}(q)$, and $P(\mathcal{C}(q) \mid q)$ denotes the posterior
 127 probability of selecting the subset $\mathcal{C}(q)$ given that q is observed to be true.

128 **Example 1.** Consider the following ProbLog program \mathcal{M} with query $q = \text{wet}$:

0.7 :: rain.	0.7 :: a.
0.7 :: sprinkler.	0.7 :: b.
0.3 :: cloudy.	0.3 :: c.
wet :- rain.	d :- a.
wet :- sprinkler.	d :- b.

129 where *rain*, *sprinkler*, *cloudy*, and *wet* are denoted by a , b , c , and d , respectively, for simplicity.

130 **Ground Probabilistic Facts of $\mathcal{M}(q)$:** $\mathcal{F}(q) = \{a, b\}$, since c is irrelevant to query d .

131 **Effective Choices:** $\mathcal{C}(q) \in \{\{\}, \{a\}, \{b\}, \{a, b\}\}$. For $\mathcal{C}(q) = \{a, b\}$, $P(\mathcal{C}(q)) = 0.7 \times 0.7 = 0.49$.
 132 **MPE Inference:** $\text{MPE}(q \mid \mathcal{M}) = \{a, b\}$, $P(\text{MPE}(q \mid \mathcal{M})) = 0.49$ and $\text{MPE}(\neg q \mid \mathcal{M}) = \{\}$,
 133 $P(\text{MPE}(\neg q \mid \mathcal{M})) = 0.09$.

134 4 Probabilistic Model Reconciliation

135 We now present our framework for generating explanations in PLP. Intuitively, it enables us to
 136 resolve discrepancies between an agent's and a human's probabilistic models, caused by incomplete
 137 information, conflicting assumptions, or differing knowledge. Model reconciliation identifies and
 138 explains these differences, fostering a shared understanding between the agent and the human.

139 4.1 Problem Settings and Assumptions

140 We consider a setting where an agent and a human user each maintain their own ProbLog programs:
 141 the agent's model \mathcal{M}_a and the human's model \mathcal{M}_h .

142 **Definition 1** (Model Inconsistency). *The agent model \mathcal{M}_a and the human model \mathcal{M}_h are said to be*
 143 *inconsistent with respect to a query q if one of the following conditions holds:*

- 144 • **Case 1:** $P(\text{MPE}(q \mid \mathcal{M}_a)) > P(\text{MPE}(\neg q \mid \mathcal{M}_a))$, but $P(\text{MPE}(q \mid \mathcal{M}_h)) < P(\text{MPE}(\neg q \mid \mathcal{M}_h))$
- 145 • **Case 2:** $P(\text{MPE}(q \mid \mathcal{M}_a)) < P(\text{MPE}(\neg q \mid \mathcal{M}_a))$, but $P(\text{MPE}(q \mid \mathcal{M}_h)) > P(\text{MPE}(\neg q \mid \mathcal{M}_h))$

146 In both cases, the agent and the human assign opposite preferences to q and $\neg q$, indicating a divergence
 147 in belief that motivates the need for model reconciliation.

148 4.2 Problem Formulation

149 To resolve the inconsistency between the agent and the human user, the agent must generate an
 150 explanation that allows the human to reconcile their model with that of the agent. To this end, we
 151 propose a logic-based formulation of model reconciliation within the ProbLog framework, referred to
 152 as a *P-MRP Explanation*. A P-MRP Explanation is formally defined as follows:

153 **Definition 2** (P-MRP Explanation). *Given that the agent model \mathcal{M}_a and the human model \mathcal{M}_h are*
 154 *inconsistent with respect to query q (as defined in Definition 1), we define $\epsilon = \langle \epsilon^+, \epsilon^- \rangle$ as a P-MRP*
 155 *explanation for q from \mathcal{M}_a to \mathcal{M}_h if and only if $\epsilon^+ \subseteq \mathcal{M}_a$, $\epsilon^- \subseteq \mathcal{M}_h$, and the updated human*
 156 *model $\mathcal{M}_h^* = (\mathcal{M}_h \cup \epsilon^+) \setminus \epsilon^-$ is both valid and consistent with \mathcal{M}_a with respect to the query q .*

157 When the human model \mathcal{M}_h is updated using a P-MRP explanation ϵ , new formulae ϵ^+ (including
 158 facts and rules) from \mathcal{M}_a are added, and formulae ϵ^- from \mathcal{M}_h are removed to ensure consistency.

159 To evaluate the quality of an explanation, we associate a cost with each candidate explanation
 160 $\epsilon = \langle \epsilon^+, \epsilon^- \rangle$, quantified by a cost function $\text{cost}(\epsilon)$ that, at a high level, reflects the effort needed by
 161 the human to incorporate the explanation. This function serves as the optimization objective and is
 162 defined as follows.

163 **Definition 3** (Explanation Cost). *Given an explanation $\epsilon = \langle \epsilon^+, \epsilon^- \rangle$, let ϵ_{fact}^+ and ϵ_{fact}^- denote the*
 164 *sets of probabilistic facts, and ϵ_{rule}^+ and ϵ_{rule}^- denote the sets of rules in ϵ^+ and ϵ^- , respectively. We*
 165 *consider the following types of modification:*²

- 166 • **Change-probability** (c_p): A cost c_p is incurred for each fact $f_i \in \epsilon_{\text{fact}}^+ \cap \epsilon_{\text{fact}}^-$, representing a
 167 probability update.
- 168 • **Add-fact** (c_f^+): A cost c_f^+ is incurred for each new fact $f_i \in \epsilon_{\text{fact}}^+ \setminus \epsilon_{\text{fact}}^-$.
- 169 • **Add-rule** (c_r^+): A cost c_r^+ is incurred for each rule $r \in \epsilon_{\text{rule}}^+$ added to the model.
- 170 • **Delete-rule** (c_r^-): A cost c_r^- is incurred for each rule $r \in \epsilon_{\text{rule}}^-$ removed from the model.

171 The total explanation cost is given by:

$$\text{cost}(\epsilon) = c_p \cdot |\epsilon_{\text{fact}}^+ \cap \epsilon_{\text{fact}}^-| + c_f^+ \cdot |\epsilon_{\text{fact}}^+ \setminus \epsilon_{\text{fact}}^-| + c_r^+ \cdot |\epsilon_{\text{rule}}^+| + c_r^- \cdot |\epsilon_{\text{rule}}^-|. \quad (3)$$

172 The task of explanation generation can be formulated as an optimization problem, defined as follows.

²We omit **delete-fact** since it is identical to **change-probability** that sets the probability of the fact to 0.

173 **Definition 4** (Optimal Explanation). Let $\mathcal{E}_{\text{valid}}$ be the set of all valid P-MRP explanations ϵ as defined
 174 in Definition 2. Then, an optimal explanation is defined as: $\epsilon^* = \operatorname{argmin}_{\epsilon \in \mathcal{E}_{\text{valid}}} \operatorname{cost}(\epsilon)$.

175 **Example 2.** Continuing the scenario in Example 1, consider the following two models \mathcal{M}_a and \mathcal{M}_h .
 176

$$\begin{array}{ll} & 0.7 :: a. \\ & 0.7 :: b. \\ \mathcal{M}_a : & 0.3 :: c. \quad \mathcal{M}_h : \quad 0.3 :: a. \\ & d : -a. \quad d : -a. \\ & d : -b. \end{array}$$

177 Let the query be d . As shown in Example 1, we have $P(\text{MPE}(d \mid \mathcal{M}_a)) > P(\text{MPE}(\neg d \mid \mathcal{M}_a))$, but
 178 $P(\text{MPE}(d \mid \mathcal{M}_h)) = P(\{a\}) = 0.3 < P(\text{MPE}(\neg d \mid \mathcal{M}_h)) = P(\emptyset) = 0.7$.

179 The following set constitutes the set of valid P-MRP explanations for d from \mathcal{M}_a to \mathcal{M}_h : $\mathcal{E}_{\text{valid}} =$
 180 $\{\langle \{0.7 :: a.\}, \{0.3 :: a.\} \rangle, \langle \{0.7 :: b., c :- b.\}, \emptyset \rangle\}$. Given a modification cost of 1 per change, the
 181 optimal explanation is: $\epsilon^* = \langle \{0.7 :: a.\}, \{0.3 :: a.\} \rangle$, where $\operatorname{cost}(\epsilon^*) = 1$. This is a **change-**
 182 **probability** action, adjusting the probability of fact a from 0.7 to 0.3.

183 5 Search-Based Explanation Generation

184 To solve the optimization problem in Definition 4, we propose two search-based algorithms for
 185 generating P-MRP explanations. The first is a generic search algorithm that serves as a baseline by
 186 exhaustively exploring all possible explanations to find a valid, cost-optimal one. While complete, it
 187 can be computationally expensive if the action spaces are large. To overcome this, we introduce an
 188 optimized search algorithm that uses pruning and cost-guided strategies to enhance efficiency.

189 Both algorithms construct explanations by incrementally modifying the human model. Actions are
 190 chosen from a two-level space: first, the type of modification (e.g., adding a fact or rule); second, the
 191 specific element to modify.

192 To formalize this process, we define the key notations of the agent and human models. Let q be
 193 a query, and let $\mathcal{M}_a(q)$ and $\mathcal{M}_h(q)$ denote the relevant ground programs under the agent model
 194 \mathcal{M}_a and the initial human model \mathcal{M}_h , respectively. The human model at timestep t is denoted by
 195 $\mathcal{M}_{h,t}$, where $\mathcal{M}_{h,0} = \mathcal{M}_h$. Let \mathcal{F}_a and $\mathcal{F}_{h,t}$ denote the sets of ground probabilistic facts in the
 196 agent model and the human model at timestep t . Similarly, let $\mathcal{F}_a(q)$ and $\mathcal{F}_{h,t}(q)$ represent the
 197 ground probabilistic facts appearing in the relevant programs $\mathcal{M}_a(q)$ and $\mathcal{M}_{h,t}(q)$, and let $\mathcal{R}_a(q)$
 198 and $\mathcal{R}_{h,t}(q)$ denote the corresponding sets of rules. For any ground fact f , we use $P_a(f)$ and $P_{h,t}(f)$
 199 to denote its probability in the agent and human models, respectively.

200 5.1 Generic Search Algorithm

201 The *generic search* algorithm exhaustively explores the explanation space without pruning. The
 202 first-level action space consists of four types of model modification operations, defined as:

$$\mathcal{A}_{\text{type}} = \{\text{change-probability}, \text{add-fact}, \text{add-rule}, \text{delete-rule}\}. \quad (4)$$

203 Given a selected action type $a_t \in \mathcal{A}_{\text{type}}$ at timestep t , the second-level action space specifies the
 204 candidate elements applicable under a_t :

- 205 • If $a_t = \text{change-probability}$, then the candidate space \mathcal{A}_t^c contains shared facts with different
 206 probabilities: $\mathcal{A}_t^c = \{f \mid f \in \mathcal{F}_a \cap \mathcal{F}_{h,t}(q), P_a(f) \neq P_{h,t}(f)\}$.
- 207 • If $a_t = \text{add-fact}$, then the candidate space is $\mathcal{A}_t^a = \mathcal{F}_a(q) \setminus \mathcal{F}_{h,t}$, representing facts available in
 208 the agent model but absent from the human model.
- 209 • If $a_t = \text{add-rule}$, then the candidate space is $\mathcal{A}_t^{r,+} = \mathcal{R}_a(q) \setminus \mathcal{R}_{h,t}(q)$, containing rules that can
 210 be added to the human model.
- 211 • If $a_t = \text{delete-rule}$, then the candidate space is $\mathcal{A}_t^{r,-} = \mathcal{R}_{h,t}(q)$, consisting of rules in the human
 212 model that can be removed. Note that we do not consider deleting rules that were previously added.

213 At each timestep t , the explanation is represented as $\epsilon_t = \langle \epsilon_t^+, \epsilon_t^- \rangle$, where $\epsilon_0^+ = \epsilon_0^- = \emptyset$ initially. The
 214 explanation is updated based on the selected action a_t and element e_t as follows:

$$\langle \epsilon_t^+, \epsilon_t^- \rangle = \begin{cases} \langle \epsilon_{t-1}^+ \cup \{P_a(e_t) :: e_t\}, \epsilon_{t-1}^- \cup \{P_{h,t}(e_t) :: e_t\} \rangle & a_t = \text{change-probability}, \\ \langle \epsilon_{t-1}^+ \cup \{P_a(e_t) :: e_t\}, \epsilon_{t-1}^- \rangle & a_t = \text{add-fact}, \\ \langle \epsilon_{t-1}^+ \cup \{e_t\}, \epsilon_{t-1}^- \rangle & a_t = \text{add-rule}, \\ \langle \epsilon_{t-1}^+, \epsilon_{t-1}^- \cup \{e_t\} \rangle & a_t = \text{delete-rule}. \end{cases} \quad (5)$$

215 After each step, the human model is updated by: $\mathcal{M}_{h,t+1} = (\mathcal{M}_h \cup \epsilon_t^+) \setminus \epsilon_t^-$.
 216 The algorithm performs a complete traversal of the explanation space (e.g., using BFS, DFS, or A*),
 217 identifying all valid explanations under which the updated human model $\mathcal{M}_{h,t}$ becomes consistent
 218 with the agent model \mathcal{M}_a with respect to the query q (see Definition 1). Among these, it returns the
 219 cost-optimal explanation (see Definition 4).

220 To formally guarantee the correctness of this approach, we present the following validity theorem
 221 and the corresponding proof is provided in Appendix A.1.

222 **Theorem 1** (Validity Guarantee). *Given agent and human models \mathcal{M}_a and \mathcal{M}_h that are inconsistent*
 223 *with respect to a query q , the search procedure described above is guaranteed to find at least one*
 224 *valid explanation $\epsilon = \langle \epsilon^+, \epsilon^- \rangle$ such that the updated human model $\mathcal{M}_h^* = (\mathcal{M}_h \cup \epsilon^+) \setminus \epsilon^-$ is*
 225 *consistent with \mathcal{M}_a regarding q .*

226 5.2 Optimized Search Algorithm

227 While the generic search algorithm is complete, it is often inefficient due to the large explanation
 228 space, where many actions are unnecessary for resolving the model inconsistency.

229 To enhance scalability, we introduce an *optimized search* algorithm that prunes irrelevant actions
 230 by focusing only on those needed to resolve the specific inconsistency. This approach identifies the
 231 minimal set of actions required, significantly reducing the search space without losing completeness.

232 To formalize this idea, we first introduce several definitions grounded in the ProbLog framework.
 233 These definitions provide the foundation for a pruning theorem and its proof, enabling precise
 234 reasoning about how model updates affect query outcomes.

235 **Definition 5** (DNF Representation of a Query). *Given a ProbLog program \mathcal{M} and a query q , let*
 236 *$\mathcal{F}(q) = \{f_1, f_2, \dots, f_n\}$ denote the set of probabilistic ground atoms in $\mathcal{M}(q)$. According to the*
 237 *semantics of ProbLog, the query q can be represented as a disjunctive normal form (DNF) formula:*

$$q = \bigvee_{i=1}^m r_i, \quad \text{where} \quad r_i = \bigwedge_{j=1}^{k_i} a_i^j.$$

238 *Each clause r_i corresponds to a derivation of q and is expressed as a conjunction of literals. Each*
 239 *literal a_i^j is either a ground atom or its negation, i.e., $a_i^j \in \mathcal{F}(q) \cup \{\neg f \mid f \in \mathcal{F}(q)\}$.*

240 This representation clarifies that satisfying any single conjunction r_i is sufficient for q to hold. Based
 241 on this structure, we now present the following theorem, which characterizes the relationship between
 242 the MPE probabilities and the DNF representation of the query.

243 **Theorem 2.** *Let \mathcal{M} be a ProbLog program and q a query with DNF representation $q = \bigvee_{i=1}^m r_i$,*
 244 *where $r_i = \bigwedge_{j=1}^{k_i} a_i^j$. Then:*

- 245 • **Case 1:** $P(\text{MPE}(q \mid \mathcal{M})) \geq P(\text{MPE}(\neg q \mid \mathcal{M})) \iff \exists i \in [m], \forall j \in [k_i], P(a_i^j) \geq 0.5$.
- 246 • **Case 2:** $P(\text{MPE}(q \mid \mathcal{M})) \leq P(\text{MPE}(\neg q \mid \mathcal{M})) \iff \forall i \in [m], \exists j \in [k_i], P(a_i^j) \leq 0.5$.

247 *Proof Sketch.* We present a proof sketch for **Case 1**, noting that the proof for **Case 2** proceeds
 248 analogously. Full details are provided in Appendix A.2.

249 (\Rightarrow) *By contradiction:* Suppose that q is more probable than $\neg q$ under MPE, yet each clause in its
 250 DNF contains a literal with a probability below 0.5. Flipping any such literal would yield a more
 251 probable explanation favoring $\neg q$, contradicting the assumption that the MPE favors q .

252 (\Leftarrow) If there exists a clause in the DNF of q such that all its literals have probability at least 0.5, then
 253 flipping them to true in the MPE of $\neg q$ results in a more probable explanation that satisfies q . \square

254 Based on Theorem 2, we can narrow the explanation search space by focusing on literals or clauses
 255 that are critical for switching the model's preference between $\neg q$ and q . This allows us to exclude
 256 actions irrelevant to belief change. We now refine the action space for each case in Definition 1.

257 **Case 1:** The agent prefers q while the human prefers $\neg q$, i.e.,

$$P(\text{MPE}(q \mid \mathcal{M}_a)) > P(\text{MPE}(\neg q \mid \mathcal{M}_a)), \quad P(\text{MPE}(q \mid \mathcal{M}_h)) < P(\text{MPE}(\neg q \mid \mathcal{M}_h)).$$

258 According to Theorem 2, increasing q 's probability in \mathcal{M}_h requires strengthening at least one DNF
 259 clause r_i with all literals meeting $P(a_i^j) \geq 0.5$ in the updated model. This allows us to prune the
 260 first-level action space to:

$$\mathcal{A}_{\text{type}} = \{\text{change-probability, add-fact, add-rule}\}.$$

261 The **delete-rule** action is excluded because increasing q only requires having *one* clause with all
 262 literals meeting the probability threshold, and deleting clauses does not help achieve this.

263 Given a selected action type $a_t \in \mathcal{A}_{\text{type}}$ at timestep t , the second-level action space defines the set of
 264 applicable candidate elements:

- 265 • If $a_t = \text{change-probability}$, the candidate space is:

$$\mathcal{A}_t^c = \{f \mid f \in \mathcal{F}_a \cap \mathcal{F}_{h,t}(q), \text{sign}(P_a(f) - 0.5) \neq \text{sign}(P_{h,t}(f) - 0.5)\}, \quad (6)$$

266 where $\text{sign}(\cdot)$ returns the sign of its input, capturing facts where the agent and human disagree on
 267 belief direction.

- 268 • If $a_t = \text{add-fact}$, the candidate space remains the same as in the *generic search* algorithm.

- 269 • If $a_t = \text{add-rule}$, we first define the current human belief set \mathcal{S} based on the updated model $\mathcal{M}_{h,t}$:

$$\mathcal{S} = \{f \mid f \in \mathcal{F}_{h,t}, P_{h,t}(f) \geq 0.5\} \cup \{\neg f \mid f \in \mathcal{F}_{h,t}, P_{h,t}(f) \leq 0.5\}. \quad (7)$$

270 That is, if the human assigns a probability strictly above 0.5 to f , we include f ; if the probability is
 271 strictly below 0.5, we include its negation $\neg f$; and if $P_{h,t}(f) = 0.5$, both f and $\neg f$ are included,
 272 reflecting a state of belief indifference.

273 According to Theorem 2, addable rules r must have all body literals supported by the current belief
 274 set, i.e., $\text{body}(r) \subseteq \mathcal{S}$. Additionally, to ensure r can increase the probability, the body must include
 275 at least one literal l where $P_{h,t}(l) \neq 0.5$. Therefore, the final candidate space is:

$$\mathcal{A}_t^{r,+} = \{r \mid r \in \mathcal{R}_a(q) \setminus \mathcal{R}_{h,t}(q), \text{body}(r) \subseteq \mathcal{S}, \exists l \in \text{body}(r) \text{ s.t. } P_{h,t}(l) \neq 0.5\}.$$

276 **Case 2:** The agent prefers $\neg q$ while the human prefers q , i.e.,

$$P(\text{MPE}(q \mid \mathcal{M}_a)) < P(\text{MPE}(\neg q \mid \mathcal{M}_a)), \quad P(\text{MPE}(q \mid \mathcal{M}_h)) > P(\text{MPE}(\neg q \mid \mathcal{M}_h)).$$

277 According to Theorem 2, decreasing q 's probability in \mathcal{M}_h requires weakening each DNF clause r_i
 278 so that at least one of its literals satisfies $P(a_i^j) \leq 0.5$ in the updated model. This allows pruning the
 279 first-level action space as follows:

$$\mathcal{A}_{\text{type}} = \begin{cases} \{\text{change-probability, delete-rule}\}, & \text{if } \mathcal{R}_{h,t}(q) \neq \emptyset \\ \{\text{change-probability, add-fact, add-rule}\}, & \text{if } \mathcal{R}_{h,t}(q) = \emptyset \end{cases}$$

280 This distinction depends on whether existing rules $\mathcal{R}_{h,t}(q)$ are present. If $\mathcal{R}_{h,t}(q) \neq \emptyset$, only
 281 changing probabilities or deleting rules is effective, as each clause must include at least one literal
 282 with $P(a_i^j) \leq 0.5$. If $\mathcal{R}_{h,t}(q) = \emptyset$, new rules or facts can be introduced.

283 Given a selected action type $a_t \in \mathcal{A}_{\text{type}}$ at timestep t , the second-level action space specifies the set
 284 of applicable candidate elements under a_t .

- 285 • If $a_t = \text{change-probability}$, the candidate space is the same as in **Case 1**, as defined in Equation 6.

- 286 • If $a_t = \text{add-fact}$, the candidate space is also identical to **Case 1**.

- 287 • If $a_t = \text{add-rule}$, we define the opposing belief set as:

$$\mathcal{S}' = \{f \mid f \in \mathcal{F}_{h,t}, P_{h,t}(f) \leq 0.5\} \cup \{\neg f \mid f \in \mathcal{F}_{h,t}, P_{h,t}(f) \geq 0.5\}.$$

288 We restrict addable rules r such that at least one literal in the body of r is supported by \mathcal{S}' and
 289 contains at least one literal l such that $P_{h,t}(l) \neq 0.5$. The candidate space is thus defined as:

$$\mathcal{A}_t^{r,+} = \{r \mid r \in \mathcal{R}_a(q) \setminus \mathcal{R}_{h,t}(q), \text{body}(r) \cap \mathcal{S}' \neq \emptyset, \exists l \in \text{body}(r) \text{ s.t. } P_{h,t}(l) \neq 0.5\}.$$

- 290 • If $a_t = \text{delete-rule}$, we allow the removal of rules whose bodies are fully supported by the current
 291 belief set \mathcal{S} (as defined in Equation 7). Formally,

$$\mathcal{A}_t^{r,-} = \{r \mid r \in \mathcal{R}_{h,t}(q), \text{body}(r) \subseteq \mathcal{S}, \exists l \in \text{body}(r) \text{ s.t. } P_{h,t}(l) \neq 0.5\}.$$

292 The update of ϵ_t follows Equation 5. We employ the A* search algorithm over the explanation
 293 space to find a cost-optimal explanation ϵ^* . Each search state represents a partial explanation, and
 294 transitions are defined by valid actions in the pruned action space.

295 At each step, the algorithm expands the lowest-cost state, generating successors by applying applicable
 296 actions. The search stops when it produces a consistent human model (Definition 1), verified using
 297 Theorem 2. The first valid solution found is guaranteed to be cost-optimal.

6 Empirical Evaluations

6.1 Estimating Action Costs via Human-User Study

This study examines how an AI agent, *Blitzcrank*, explains its decisions in an intelligent warehouse, determining whether goods should be delivered. Participants compared explanation pairs and chose the one they felt best clarified the agent’s reasoning.

Data Collection. We recruited 128 participants via Prolific [21], ensuring a diverse sample.³ Participants were fluent English speakers and were compensated USD 2.00. After attention and coherence checks, data from 100 participants were retained for analysis.

Cost Estimation. To estimate the relative cognitive effort of different explanation types, we used the Bradley-Terry model [22] on the pairwise comparison data. Each action a_i had a strength parameter β_i , with higher β_i values indicating greater participant preference. The cost of an action was defined as the negative of its strength [23], $-\beta_i$, making more preferred actions correspond to lower costs. To ensure non-negative costs compatible with search algorithms (e.g., A^*), we exponentiated the negated strength values. The final cost of each action was defined $\text{cost}(a_i) = e^{-\beta_i}$. Based on this formulation, the estimated costs for the four explanation actions are shown in Table 1.

Table 1: Estimated Costs for Each Explanation Action.

action a_i	cost(a_i)
change-probability	0.9801
add-fact	0.8688
add-rule	1.0202
delete-rule	1.1511

6.2 Computational Results

This section presents the computational performance of the *Generic Search* and *Optimized Search* algorithms for model reconciliation, aiming to find the cost-optimal explanation using the costs in Table 1. Our experiments were run on a MacBook Pro (M2, 16GB RAM) machine.

Experimental Setup. Our experiments use two models: Agent and Human.

- Agent Model \mathcal{M}_a : Each \mathcal{M}_a contains $|\mathcal{F}_a| = 10, 20, 50$, or 100 probabilistic facts and $|\mathcal{R}_a| = 15, 10, 25$, or 50 logical rules, respectively, all related to the same query. Facts have randomly assigned probabilities, and rules have bodies of 2-4 literals, generated based on cases in Definition 1.
- Human Model \mathcal{M}_h : Derived from each \mathcal{M}_a at four complexity levels $l \in \{20\%, 40\%, 60\%, 80\%\}$, reflecting the percentage of probabilistic facts that differ. Each differing fact has a 1/3 chance of being: modified (probability flipped), removed, or replaced (new fact). Human model rules share the same heads as the agent model but are built using existing facts, with the rule count as: $|\mathcal{R}_h| = \lfloor |\mathcal{R}_a| \cdot (1 - 1/3 \cdot l) \rfloor$.

We generate 100 Agent-Human Model pairs for each configuration, totaling 1,600 pairs (4 Agent settings \times 4 complexity levels \times 100 repetitions) for each case in Definition 1.

Evaluation Metrics. All experiments are capped at 600 seconds per run.

- Average Time: The average runtime (in milliseconds) for runs completed within the time limit.
- Timeout Count: The number of runs that exceed the time limit for each configuration.
- Average Cost: The mean cost of optimal explanation for runs completed within the time limit.

Results and Analysis. Table 2 compares the performance of two search algorithms.

- **Impact of Model Size on Runtime and Timeout Counts:** As model size increases, runtime for both algorithms grows, but with the optimized algorithm significantly outperforming the generic search one. For example, in Case 2 with 20% complexity, *Generic Search* jumps from 36.09 ms (10 facts) to 29,758.74 ms (20 facts) with 5 timeouts. In contrast, *Optimized Search* remains stable, rising slightly from 25.41 ms to 25.70 ms. For 50 and 100 facts, *Generic Search* consistently times out, while *Optimized Search* maintains feasible runtimes (85.94 ms for 50 facts and 22,745.00 ms for 100 facts), with only 3 timeouts in the largest case.
- **Impact of Complexity on Runtime:** Higher complexity means more fact differences but fewer rules in the human model, affecting the two cases differently. In **Case 1**, runtime increases because the action space for change-probability and add-fact expands, while delete-rule actions decrease. Since delete-rule is not essential, the larger search space complicates finding optimal explanations.

³Ethics approval was obtained from our university’s IRB. The human-subject study, collected data, and implementation will be released in a public repository with the camera-ready version.

Table 2: Performance Comparison of Two Algorithms across Two Cases in Definition 1.

Agent Facts	Agent Rules	Human Complexity	Case 1						Case 2					
			Avg. Cost	Generic Search		Optimized Search		Avg. Cost	Generic Search		Optimized Search			
				Avg. (ms)	# Timeout	Avg. (ms)	# Timeout		Avg. (ms)	# Timeout	Avg. (ms)	# Timeout		
10	5	20%	1.36	28.71	0	25.45	0	1.75	36.09	0	25.41	0		
		40%	1.50	29.65	0	25.43	0	1.69	36.22	0	25.33	0		
		60%	1.95	35.76	0	25.54	0	1.47	30.75	0	26.02	0		
		80%	1.97	33.58	0	26.60	0	1.30	28.05	0	26.09	0		
20	10	20%	1.18	1,266.76	0	25.69	0	2.33	29,758.74	5	25.70	0		
		40%	1.40	1,846.70	0	26.56	0	2.17	19,290.40	1	26.45	0		
		60%	1.44	3,546.33	0	27.74	0	1.99	11,657.01	1	26.73	0		
		80%	1.53	4,406.19	0	35.03	0	1.77	3,187.11	0	26.81	0		
50	25	20%	1.04	—	100	28.55	0	5.02	—	100	85.94	0		
		40%	1.06	—	100	31.61	0	4.35	—	100	696.74	0		
		60%	1.08	—	100	40.44	0	4.21	—	100	1,126.31	0		
		80%	1.06	—	100	84.11	0	3.19	—	100	749.74	0		
100	50	20%	0.99	—	—	36.43	0	8.15	—	—	22,745.00	3		
		40%	0.99	—	—	51.58	0	6.74	—	—	68,103.09	31		
		60%	0.99	—	—	85.71	0	5.50	—	—	118,851.28	47		
		80%	1.01	—	—	195.98	0	4.32	—	—	116,862.71	55		

* Note that we do not run this case with the generic search because it consistently timed out in smaller configurations.

In **Case 2**, for *Generic Search*, higher complexity reduces runtime because fewer delete-rule actions (which are costly) allow A* to focus on lower-cost actions. In *Optimized Search* (particularly with 50 facts), runtime first increases due to more differing facts but then decreases as fewer rules reduce delete-rule actions, balancing between rule deletions and fact modifications.

- **Impact on Explanation Cost:** As model size increases, explanation cost decreases in Case 1 but increases in Case 2. In **Case 1**, a larger human model with more facts and rules lets the agent use change-probability actions, which are lower in cost. By Theorem 2, only one rule needs to meet the condition, making lower-cost explanations more likely. In **Case 2**, more rules in the human model require more costly delete-rule actions. At fixed model sizes, higher complexity has the opposite effect: it raises the cost in **Case 1** because more fact differences require alignment but lowers the cost in **Case 2** because fewer rules need deletion.

Overall, *Optimized Search* demonstrates superior scalability and robustness compared to *Generic Search*, with lower average runtimes and fewer timeouts, even in complex settings. These results underscore the importance of efficient search strategies for scalable model reconciliation.⁴

7 Conclusion and Discussion

In this paper, we present a model reconciliation framework within probabilistic logic programming (PLP). Our approach formalizes reconciliation under uncertainty using ProbLog to represent an agent’s and a human’s probabilistic models, identifying and resolving inconsistencies in MPE outcome probabilities. We introduce a cost-based explanation model that quantifies the cognitive effort of model updates, enabling the generation of cost-optimal explanations that minimally adjust the human’s model. To generate these explanations, we develop two search algorithms: a generic search algorithm and an optimized search algorithm guided by theoretical insights for pruning the search space. Our approach is validated through a user study demonstrating how different explanation types impact user understanding and a computational evaluation, where the optimized search consistently outperforms the generic method in runtime and scalability.

Our framework enhances human-AI interaction by providing clear, cost-optimal explanations for AI decisions, improving user understanding and trust across various domains. By aligning the user’s understanding with the agent’s model, it empowers informed decision-making. However, it also raises ethical concerns: flawed agent models may reinforce incorrect beliefs, and users may over-rely on AI without critical evaluation. Adaptive explanations could also manipulate user beliefs. Ensuring transparency and ethical design is essential.

Despite its strengths, our framework has several limitations. As a logic-based approach, it has been tested primarily in computational experiments and a controlled user study, leaving its effectiveness in real-world scenarios unverified. Although we account for uncertainty through probabilistic models, the human model is assumed to be consistent and calibrated, which may not always hold in practice. Finally, while the optimized search improves scalability, it still encounters timeouts in the largest settings, highlighting the need for better heuristic optimization.

⁴However, even in the largest settings, Optimized Search faces more timeouts, underscoring the need for better heuristic functions, though a query with 50 facts is typically sufficient.

References

- [1] Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. Explanations as model reconciliation-a multi-agent perspective. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 277–283, 2017.
- [2] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 156–163, 2017.
- [3] Tran Cao Son, Van Nguyen, Stylianos Loukas Vasileiou, and William Yeoh. Model reconciliation in logic programs. In *Proceedings of Logics in Artificial Intelligence (JELIA)*, pages 393–406, 2021.
- [4] Stylianos Loukas Vasileiou, Alessandro Previti, and William Yeoh. On exploiting hitting sets for model reconciliation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 6514–6521, 2021.
- [5] Daan Fierens, Guy Van den Broeck, Ingo Thon, Bernd Gutmann, and Luc De Raedt. Inference in probabilistic logic programs using weighted cnf’s. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 211–220, 2011.
- [6] Stylianos Loukas Vasileiou, William Yeoh, Tran Cao Son, and Alessandro Previti. Explanations as model reconciliation via probabilistic logical reasoning. In *Proceedings of the International Conference on Knowledge Representation and Reasoning Workshop on Explainable Logic-Based Knowledge Representation (XLoKR)*, 2021.
- [7] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Sht. Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.
- [8] Maria Fox, Derek Long, and Daniele Magazzeni. Explainable planning. *arXiv preprint arXiv:1709.10256*, 2017.
- [9] Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. Handling model uncertainty and multiplicity in explanations via model reconciliation. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 518–526, 2018.
- [10] David Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.
- [11] Peter Flach. *Simply logical - intelligent reasoning by example*. Wiley professional computing. Wiley, 1994.
- [12] Taisuke Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of International Conference on Logic Programming (ICLP)*, pages 715–729, 1995.
- [13] Taisuke Sato and Yoshitaka Kameya. Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, 15:391–454, 2001.
- [14] David L Poole. Exploiting the rule structure for decision making within the independent choice logic. *arXiv preprint arXiv:1302.4978*, 2013.
- [15] Joost Vennekens, Sofie Verbaeten, and Maurice Bruynooghe. Logic programs with annotated disjunctions. In *Proceedings of International Conference on Logic Programming (ICLP)*, pages 431–445, 2004.
- [16] Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pages 1–27. 2008.
- [17] D. Fierens, G. Van den Broeck, B. Gutmann I. Thon, and L. De Raedt. Inference in probabilistic logic programs using weighted CNF’s. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 211–220, 2011.

- 432 [18] Dimitar Shterionov, Joris Renkens, Jonas Vlasselaer, Angelika Kimmig, Wannes Meert, and
433 Gerda Janssens. The most probable explanation for probabilistic logic programs with annotated
434 disjunctions. In *Proceedings of the International Conference on Inductive Logic Programming*
435 (*ILP*), pages 139–153, 2015.
- 436 [19] Angelika Kimmig, Bart Demoen, Luc De Raedt, Vitor Santos Costa, and Ricardo Rocha. On
437 the implementation of the probabilistic logic programming language ProbLog. *Theory and*
438 *Practice of Logic Programming*, 11(2-3):235–262, 2011.
- 439 [20] Joris Renkens, Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt. Explanation-based
440 approximate weighted model counting for probabilistic logics. In *Proceedings of the AAAI*
441 *Conference on Artificial Intelligence (AAAI)*, pages 2490–2496, 2014.
- 442 [21] Stefan Palan and Christian Schitter. Prolific.ac – a subject pool for online experiments. *Journal*
443 *of Behavioral and Experimental Finance*, 17:22–27, 2018.
- 444 [22] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the
445 method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- 446 [23] Kaivalya Rawal and Himabindu Lakkaraju. Learning recourse costs from pairwise feature
447 comparisons. *arXiv preprint arXiv:2409.13940*, 2024.

A Appendix

A.1 Proof of Theorem 1

Proof. The search algorithm exhaustively explores all possible combinations of explanation components. In particular, it will consider the explanation $\epsilon = \langle \epsilon^+, \epsilon^- \rangle$ where $\epsilon^+ = \mathcal{M}_a \setminus \mathcal{M}_h$ and $\epsilon^- = \mathcal{M}_h \setminus \mathcal{M}_a$. Applying this explanation yields the updated human model:

$$\mathcal{M}_h^* = (\mathcal{M}_h \cup \epsilon^+) \setminus \epsilon^- = \mathcal{M}_a.$$

Since $\mathcal{M}_h^* = \mathcal{M}_a$, the models are fully aligned, and thus consistent with respect to q . Therefore, the existence of at least one valid explanation is guaranteed. \square

A.2 Proof of Theorem 2

Proof. We provide the proof for Case 1. The proof for Case 2 proceeds in a similar manner.

$$(\Rightarrow) P(\text{MPE}(q \mid \mathcal{M})) \geq P(\text{MPE}(\neg q \mid \mathcal{M})) \Rightarrow \exists i \in [m], \forall j \in [k_i], P(a_i^j) \geq 0.5$$

We prove by contradiction. Suppose $P(\text{MPE}(q \mid \mathcal{M})) \geq P(\text{MPE}(\neg q \mid \mathcal{M}))$ but $\forall i \in [m], \exists j \in [k_i], P(a_i^j) < 0.5$.

Let $\mathcal{C}(q) = \text{MPE}(q \mid \mathcal{M})$ be the most probable explanation under which q holds. Since $q = \bigvee_{i=1}^m r_i$, q holds under $\mathcal{C}(q)$ if and only if at least one r_i is true. Let r_1 be one such clause satisfied in $\mathcal{C}(q)$.

By assumption, in r_1 there exists some $j \in [k_1]$ such that $P(a_1^j) < 0.5$. Without loss of generality, assume $j = 1$, i.e., $P(a_1^1) < 0.5$. Construct a new explanation \mathcal{C}' by setting a_1^1 to false, and keeping all other assignments unchanged.

Since $P(a_1^1) < 0.5$, it follows that

$$P(\mathcal{C}') > P(\mathcal{C}(q)).$$

We now show that q does not hold under \mathcal{C}' . Since a_1^1 is false in \mathcal{C}' , the clause r_1 is no longer satisfied. If q were still true under \mathcal{C}' , then some other r_i must be satisfied. But this would mean that \mathcal{C}' is a valid explanation for q with higher probability than $\mathcal{C}(q)$, contradicting the definition of $\mathcal{C}(q)$ as the most probable explanation of q .

Therefore, q does not hold under \mathcal{C}' , i.e., \mathcal{C}' satisfies $\neg q$. This leads to a contradiction:

$$P(\text{MPE}(q \mid \mathcal{M})) = P(\mathcal{C}(q)) < P(\mathcal{C}') \leq P(\text{MPE}(\neg q \mid \mathcal{M})).$$

Hence, the assumption must be false.

$$(\Leftarrow) \exists i \in [m], \forall j \in [k_i], P(a_i^j) \geq 0.5 \Rightarrow P(\text{MPE}(q \mid \mathcal{M})) \geq P(\text{MPE}(\neg q \mid \mathcal{M}))$$

Suppose $\exists i \in [m], \forall j \in [k_i], P(a_i^j) \geq 0.5$. Without loss of generality, assume this holds for r_1 .

Let $\mathcal{C}(\neg q) = \text{MPE}(\neg q \mid \mathcal{M})$ be the most probable explanation under which q does not hold. Since $q = \bigvee_{i=1}^m r_i$, $\neg q$ holds under $\mathcal{C}(\neg q)$ only if all r_i are false. In particular, r_1 must be false under $\mathcal{C}(\neg q)$, meaning that at least one literal in r_1 is assigned false. Let

$$A = \{a_1^j \mid a_1^j \text{ is false in } \mathcal{C}(\neg q), j \in [k_1]\}.$$

Construct a new explanation \mathcal{C}' by flipping the truth values of all $a_1^j \in A$ to true, and keeping all other assignments unchanged.

Since each $P(a_1^j) \geq 0.5$, this modification leads to:

$$P(\mathcal{C}') \geq P(\mathcal{C}(\neg q)).$$

Furthermore, since $r_1 = \bigwedge_{j=1}^{k_1} a_1^j$ and all a_1^j are now true in \mathcal{C}' , we have that r_1 is satisfied in \mathcal{C}' , and hence q holds.

This yields:

$$P(\text{MPE}(\neg q \mid \mathcal{M})) = P(\mathcal{C}(\neg q)) \leq P(\mathcal{C}') \leq P(\text{MPE}(q \mid \mathcal{M})).$$

\square

B Supplemental Materials: Search Algorithm for Explanation Generation

This section provides the full pseudocode and a comparative example to illustrate how the search algorithm operates under different explanation settings.

B.1 Pseudocode

This section presents the A*-based search algorithm used for generating cost-optimal explanations. The algorithm can operate in two modes:

- **Generic:** no pruning; full action space explored.
- **Optimized:** case-specific pruning based on model inconsistency.

Algorithm 1 A*-Based Search Algorithm for Explanation Generation

Require: Human model \mathcal{M}_h , Agent model \mathcal{M}_a , Query q , Case $\in \{1, 2\}$, Prune $\in \{\text{True}, \text{False}\}$
Ensure: Cost-optimal explanation $\epsilon^* = \langle \epsilon^+, \epsilon^- \rangle$

- 1: Initialize priority queue $\mathcal{Q} \leftarrow \{(\langle \emptyset, \emptyset \rangle, 0)\}$
- 2: Initialize visited set $\mathcal{V} \leftarrow \emptyset$
- 3: **while** \mathcal{Q} is not empty **do**
- 4: Pop explanation $\epsilon_t = \langle \epsilon_t^+, \epsilon_t^- \rangle$ with lowest $f = g + h$
- 5: $\mathcal{M}_{h,t} \leftarrow (\mathcal{M}_h \cup \epsilon_t^+) \setminus \epsilon_t^-$
- 6: **if** $\text{IsConsistent}(\mathcal{M}_{h,t}, \mathcal{M}_a, q, \text{Case}, \text{Prune})$ **then**
- 7: **return** ϵ_t
- 8: **end if**
- 9: **if** $\epsilon_t \in \mathcal{V}$ **then**
- 10: **continue**
- 11: **end if**
- 12: Add ϵ_t to \mathcal{V}
- 13: Determine $\mathcal{A}_{\text{type}}$ (pruned or full) based on Prune and Case
- 14: **for all** action type $a_t \in \mathcal{A}_{\text{type}}$ **do**
- 15: Compute candidate set $\mathcal{A}_t^{a_t}$ (pruned or full)
- 16: **for all** $e_t \in \mathcal{A}_t^{a_t}$ **do**
- 17: $\epsilon_{t+1} \leftarrow \text{Apply}(a_t, e_t, \epsilon_t)$
- 18: $g \leftarrow \text{Cost}(\epsilon_{t+1}), h \leftarrow \text{Heuristic}(\epsilon_{t+1}), f \leftarrow g + h$
- 19: Insert ϵ_{t+1} into \mathcal{Q} with priority f
- 20: **end for**
- 21: **end for**
- 22: **end while**
- 23: **function** $\text{ISCONSISTENT}(\mathcal{M}_{h,t}, \mathcal{M}_a, q, \text{Case}, \text{Prune})$
- 24: **if not** Prune **then** ▷ Generic Search
- 25: **if** Case = 1 **then**
- 26: **return** $P(\text{MPE}(q \mid \mathcal{M}_{h,t})) > P(\text{MPE}(\neg q \mid \mathcal{M}_{h,t}))$
- 27: **else if** Case = 2 **then**
- 28: **return** $P(\text{MPE}(q \mid \mathcal{M}_{h,t})) < P(\text{MPE}(\neg q \mid \mathcal{M}_{h,t}))$
- 29: **end if**
- 30: **else if** Prune **then** ▷ Optimized Search
- 31: **if** Case = 1 **then**
- 32: **return** $\exists i \in [m], \forall j \in [k_i], P(a_i^j) \geq 0.5$
- 33: **else if** Case = 2 **then**
- 34: **return** $\forall i \in [m], \exists j \in [k_i], P(a_i^j) \leq 0.5$
- 35: **end if**
- 36: **end if**
- 37: **end function**

The Search Algorithm follows the A* search paradigm. It evaluates all possible explanation actions at each step, using a simple admissible heuristic:

$$\text{Heuristic}(\epsilon_t) = \min(c_p, c_f^+, c_r^+, c_r^-),$$

494 which estimates the minimal cost needed to reach consistency from the current explanation state ϵ_t .
 495 Since it is a lower bound on the actual remaining cost, the heuristic is admissible under A* semantics.

496 B.2 Illustrative Example: Action Space Differences

497 To demonstrate how action space pruning affects search behavior, we compare the sets of candidate
 498 actions considered by the Generic and Optimized Search Algorithms under both Case 1 and Case 2.
 499 For each case, we present the initial agent and human models, followed by the corresponding action
 500 types and candidate elements at the *first* search step.

501 **Case 1** This example illustrates a scenario in which the agent model supports the query d , while the
 502 human model does not. The goal is to strengthen the human model so that d becomes more probable.

$$\begin{array}{ll} & 0.1 :: a. \\ & 0.6 :: b. \\ & 0.7 :: c. \\ \mathcal{M}_a : & 0.8 :: e. \\ & d : -c, e. \\ & d : -a, b. \\ & d : -a, c. \end{array} \quad \begin{array}{l} 0.2 :: a. \\ 0.2 :: b. \\ 0.3 :: c. \\ d : -b. \\ d : -b, c. \end{array}$$

503 *Generic Search Algorithm.* First-level action types:

$$\mathcal{A}_{\text{type}} = \{\text{change-probability, add-fact, add-rule, delete-rule}\}$$

504 Candidate sets:

- 505 • $\mathcal{A}_t^c = \{a, b, c\}$ (shared facts with differing probabilities)
- 506 • $\mathcal{A}_t^a = \{e\}$ (fact e exists in \mathcal{M}_a but not in \mathcal{M}_h)
- 507 • $\mathcal{A}_t^{r,+} = \{d : -a, b., d : -a, c.\}$
- 508 • $\mathcal{A}_t^{r,-} = \{d : -b., d : -b, c.\}$

509 *Optimized Search Algorithm.* First-level action types:

$$\mathcal{A}_{\text{type}} = \{\text{change-probability, add-fact, add-rule}\} \quad (\text{delete-rule is pruned})$$

510 Pruned candidate sets:

- 511 • $\mathcal{A}_t^c = \{b, c\}$ (since $P_a(b) = 0.6 > 0.5$, $P_h(b) = 0.2 < 0.5$, and $P_a(c) = 0.7 > 0.5$,
 512 $P_h(c) = 0.3 < 0.5$)
- 513 • $\mathcal{A}_t^a = \{e\}$ (same as Generic)
- 514 • $\mathcal{A}_t^{r,+} = \emptyset$ (current belief set $\mathcal{S} = \{\neg a, \neg b, \neg c\}$ from Eq. (7) supports no rule)

515 **Case 2** In contrast, this example is symmetric to Case 1, with the agent and human models swapped.
 516 The human model supports d , while the agent does not. The goal is to weaken the human model's
 517 belief in d .

$$\begin{array}{ll} & 0.1 :: a. \\ & 0.2 :: b. \\ & 0.2 :: b. \\ \mathcal{M}_a : & 0.3 :: c. \\ & d : -b. \\ & d : -b, c. \end{array} \quad \begin{array}{l} 0.6 :: b. \\ 0.7 :: c. \\ 0.8 :: e. \\ d : -c, e. \\ d : -a, b. \\ d : -a, c. \end{array}$$

518 *Generic Search Algorithm.* First-level action types:

$$\mathcal{A}_{\text{type}} = \{\text{change-probability, add-fact, add-rule, delete-rule}\}$$

519 Candidate sets:

520 • $\mathcal{A}_t^c = \{a, b, c\}$ (shared facts with differing probabilities)

521 • $\mathcal{A}_t^a = \emptyset$

522 • $\mathcal{A}_t^{r,+} = \{d : -b., d : -b, c.\}$

523 • $\mathcal{A}_t^{r,-} = \{d : -a, b., d : -a, c., d : -c, e.\}$

524 *Optimized Search Algorithm.* Since $\mathcal{R}_{h,t}(q) \neq \emptyset$, the pruned first-level action types are:

$$\mathcal{A}_{\text{type}} = \{\text{change-probability, delete-rule}\}$$

525 Pruned candidate sets:

526 • $\mathcal{A}_t^c = \{b, c\}$ (as $P_a(b) = 0.2 < 0.5$, $P_h(b) = 0.6 > 0.5$, and similarly for c)

527 • $\mathcal{A}_t^{r,-} = \{d : -c, e.\}$ (based on current belief set $\mathcal{S} = \{-a, b, c, e\}$)

528 In both cases, pruning drastically reduces the number of candidate actions without compromising
 529 the final outcome. This validates the theoretical pruning conditions derived in Theorem 2 and
 530 demonstrates their practical utility in guiding efficient explanation search.