Editable XAI: Toward Bidirectional Human-AI Alignment with Co-Editable Explanations of Interpretable Attributes

Haoyang Chen
chen_haoyang@u.nus.edu
Department of Computer Science
National University of Singapore
Singapore

Jingwen Bai
jingwenbai@u.nus.edu
Department of Computer Science
National University of Singapore
Singapore

Tian Fang
tfang@u.nus.edu
College of Design and Engineering
National University of Singapore
Singapore

Brian Y. Lim*
brianlim@nus.edu.sg
Department of Computer Science
National University of Singapore
Singapore

ABSTRACT

While Explainable AI (XAI) helps users understand AI decisions, misalignment in domain knowledge can lead to disagreement. This inconsistency hinders understanding, and because explanations are often read-only, users lack the control to improve alignment. We propose making XAI editable, allowing users to write rules to improve control and gain deeper understanding through the generation effect of active learning. We developed CoExplain, leveraging a neural network for universal representation and symbolic rules for intuitive reasoning on interpretable attributes. CoExplain explains the neural network with a faithful proxy decision tree, parses user-written rules as an equivalent neural network graph, and collaboratively optimizes the decision tree. In a user study (N=43), CoExplain and manually editable XAI improved user understanding and model alignment compared to read-only XAI. CoExplain was easier to use with fewer edits and less time. This work contributes Editable XAI for bidirectional AI alignment, improving understanding and control.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Human-centered computing → Empirical studies in HCI**.

KEYWORDS

Explainable AI, human-AI alignment, interactive machine learning

ACM Reference Format:
Haoyang Chen, Jingwen Bai, Tian Fang, and Brian Y. Lim. 2026. Editable XAI: Toward Bidirectional Human-AI Alignment with Co-Editable Explanations of Interpretable Attributes. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 37 pages. https://doi.org/10.1145/3772318.3791375

1 INTRODUCTION

Explainable AI (XAI) helps users understand the reasoning behind AI decisions. However, when the AI is incorrect or incongruent with a user's domain knowledge or beliefs, this results in human–AI misalignment, which can cause distrust, errors, and extra work for users to verify or override AI decisions [64, 130]. XAI can also exacerbate the problem: by presenting explanations that appear reasonable, it can reinforce automation bias, where users place undue trust in AI outputs and overlook potential errors [20]. When explanations are static and read-only, misalignment persists, leaving users unable to correct errors or incorporate their own knowledge.

We propose **Editable XAI** and hypothesize that allowing users to rewrite the rules of AI reasoning can improve alignment. By allowing users and AI to collaboratively inspect and revise a shared medium explanation, users can better understand the AI behavior and identify the misalignment. Beyond improving alignment, the act of editing explanations engages users in actively producing content rather than passively consuming it, leveraging the *generation effect* from active learning [115, 125]. In this way, editable explanations promote bi-directional alignment [112, 113]: I) users can adapt AI reasoning to their domain knowledge, II) while simultaneously deepening their own understanding of how the AI makes decisions.

To explore this idea, we conducted an elicitation study to understand how users may want to edit or override AI reasoning. From this study, we identified several design needs for **Editable XAI**. First, explanations should be contextualized in a *writable format*, such as rules, so that users can directly adjust rather than only read them. Second, systems should support *user-written* rules that capture domain knowledge with user familiar terms, rather than relying solely on data-driven patterns. Third, AI assistance should provide *automated enhancements* to user rules, such as adjusting thresholds or reorganizing rule topology, to help close knowledge gaps. Finally, automated edits should be *constrained* to remain consistent with user intent and avoid introducing unnecessary complexity. These findings directly informed the design of our system.

Based on these requirements, we introduce **CoExplain**, a framework for **Editable XAI** that supports three modes of interaction: **Read**, **Write**, and **Enhance**. We use decision tree rules as the interaction medium as they are human-interpretable and human-editable while compatible with machine learning models. We use neural

*Corresponding author

CHI '26, April 13–17, 2026, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3791375

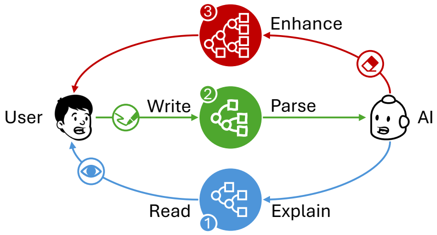


Figure 1: The three interaction modes for Editable XAI. (1) Read: users inspect explanations generated by the AI, (2) Write: users modify explanations to guide the AI, and (3) Enhance: users and the AI collaboratively refine explanations. Creating a bidirectional alignment between the user and AI.

networks as the underlying AI model to exploit its expressivity and flexible training with gradient descent. For **Read**, CoExplain distills a neural network into a decision tree to provide faithful and accessible explanations of model decisions. For **Write**, user-authored rules are parsed into an equivalent neural network, enabling the system to incorporate domain knowledge while maintaining trainability. For **Enhance**, CoExplain augments user-written rules in two ways: conservatively refining threshold values to adjust decision boundaries, and more aggressively reorganizing the tree's topology to improve predictive performance. These actions are linked through neurosymbolic approaches that maintain equivalence between decision trees and neural networks via distillation, parsing, backpropagation, and regularization, thereby enabling bi-directional collaboration between humans and AI.

We evaluated CoExplain in a user study with 43 participants, compared it with Read-only XAI and manually Editable XAI without AI enhancements. Both editable techniques improved understanding and alignment compared to read-only explanations. Qualitative findings showed that writing rules helped users align AI reasoning with their own knowledge and deepened their understanding on model behavior, while AI enhancements were valued for refining thresholds and restructuring rules under user control. Quantitative results showed that both Editable and CoExplain improved understanding, with CoExplain striking a balance between maintaining alignment with users' initial knowledge and achieving near-optimal model performance. CoExplain also reduced editing effort by providing on-demand enhancements, saving users time on refining thresholds and topology. Participants generally perceived it as easier to use and appreciated its collaborative nature.

In summary, this paper makes two **contributions**:

1) Insights on how users edit explanatory rules, both independently and collaboratively with AI, highlighting needs for **Editable XAI**, AI-guided enhancements, and constrained automation.
2) **CoExplain**, a neurosymbolic framework and interactive XAI tool that supports read, write, and enhance rule-based explanation, enabling bi-directional human-AI alignment.

Our evaluations showed that editable explanations, CoExplain in particular, foster more effective and efficient human–AI alignment than static, read-only explanations. We discuss the scope, limitations, and generalization of Editable XAI, and outline future directions toward effective human–AI collaboration and alignment.

2 RELATED WORK

We discuss related work on how explainable AI is mostly used for reading only, interactive machine learning methods for human-AI alignment, neurosymbolic methods to interface semantic rules with expressive AI models and end-user development to increase access and control of smart systems. Prior works in these areas are mostly one directional in solving the misalignment problem, while Editable XAI approach it in a bi-directional way by deepening both the user's understanding of AI and the AI's alignment with user.

2.1 Usage of Explainable Artificial Intelligence

Explainable AI (XAI) has been a key research area to increase transparency in machine learning systems. Most existing approaches focus on generating post-hoc justifications for model outputs, such as saliency maps [1, 2, 55, 111] or gradient-based attributions [4, 5, 118], primarily targeting technical audiences. These methods often leave non-expert users underserved, particularly in high-stake domains where understanding both the structure and reasoning of AI systems is crucial [75, 129]. Such post-hoc explanations are **one-way communications** that show users what the model did, rather than inviting them to question or modify. Even seemingly user-friendly formats like natural language rationalization [35] or example-based explanations [11, 23, 66, 68] limit interactivity and rarely expose the underlying logical structure of the model.

Recent work emphasizes the importance of interactive explanations, where users are not just recipients of reasoning but active participants in shaping it [79, 116]. By allowing users to modify explanations, systems can support deeper understanding [49, 78], however, these works treat user input as optional or post-hoc, without letting user input affect the AI's reasoning process. On the other hand, **Editable XAI** includes users as **active collaborators** with the AI system, rather than as passive receivers of the static explanations, treating user input as a core part of model reasoning and refinement. By allowing users to **directly inspect and modify** the AI's reasoning with decision tree explanations [12, 26], **CoExplain** creates a **bi-directional alignment** between the user and the AI.

2.2 Interactive Machine Learning for Human–AI Alignment

While XAI improves transparency, aligning AI systems with user expectations often requires **interactive interventions** rather than passive explanations. Interactive Machine Learning (IML) places users in the loop to adjust models, leveraging domain knowledge and improving alignment under scarce or noisy data [36, 98].

Most prior IML systems focus on **data-level feedback**, such as labeling instances [10, 22], modifying features [17, 126], or curating datasets [9, 102]. [114] explored using gradient to correct specific mistakes, but cannot inspect and correct the model's reasoning as a whole. Such instance-based corrections are **local**, requiring users to inspect examples individually and making global misalignment patterns difficult to detect.

In contrast, **model-level interactions**, where users directly shape a model's structure or reasoning logic, remain underexplored [98]. Recent work using rules or decision trees offers interpretable interfaces [31, 49, 72, 73, 102], but these are typically post-hoc aids rather than manipulable model components.

Editable XAI advances this space by treating explanations as **both interpretable and editable substrates**. Users can inspect and revise rules that govern **model-wide behavior**, creating a collaborative, continuous feedback loop. This enables **global alignment**, correcting reasoning patterns rather than isolated instances, making explanations not only clarify the model's logic but also serve as a manipulable substrate for alignment.

2.3 Integrating Human Knowledge in Neural Networks with Neuro-Symbolic Learning

Neural-symbolic learning bridges human reasoning and statistical learning by integrating symbolic rules into neural models [34, 40]. These hybrid approaches enhance interpretability and allow networks to incorporate logical constraints, aligning model behavior more closely with human expectations.

Prior work typically embeds rules through **formal logic encodings** [7, 29, 106, 127] or **constraint-based objectives**, such as differentiable logic or regularization [59, 82]. While effective, these methods demand technical expertise to specify rules and integrate them into training pipelines, limiting access for domain experts or end-users whose knowledge is critical for meaningful alignment.

This gap highlights the need for **user-accessible** neuro-symbolic methods that allow **intuitive knowledge input** and **iterative refinement**. By enabling symbolic rules to be directly editable and interpretable, **Editable XAI** supports collaborative human–AI reasoning, allowing users to guide model behavior without requiring technical expertise. This shifts Neural-symbolic methods from technically encoded rules to **human-centered, editable knowledge**.

2.4 End-User Development for Editable Systems

Many real-world applications involve **non-technical users** who possess domain knowledge but lack effective means to integrate it into AI systems. **End-User Development (EUD)** addresses this gap by enabling users to modify system behavior without programming expertise [37, 85]. Rule-based interfaces are common in EUD as they provide a **transparent representation of user intent** [8, 32]. Prior work demonstrates EUD applications in personalizing smart environments, IoT behaviors, and adaptive systems [44, 48, 56].

However, these efforts generally focus on **simple if–then rules**, which limits their applicability to complex decision structures or integration with AI reasoning. Extending EUD to support **richer rule authoring and editing** would allow end-users to contribute domain knowledge in a way that directly influences model behavior, while maintaining accessibility and interpretability. This perspective positions EUD not just as a personalization tool, but as a potential bridge for **human-guided, editable AI systems**.

3 ELICITATION USER STUDY

Despite advancements in tools for rule editing, end-user programming, and interactive machine learning [31, 44, 49], a significant gap exists in the support for editable AI explanations. To address this, we developed a basic rule editor tool informed by Tree Edit Distance (TED, Zhang and Shasha [128]) to include tree editing operations for inserting, deleting, and updating rules. We then used this tool as a *design probe* in a user study to elicit user needs for automatic and AI-supported editing of XAI explanations.

3.1 Method

3.1.1 Probe apparatus and user task. This study was conducted using a basic editing tool we developed with an attribute list where users can select the attributes to use and a canvas where they can edit their rule with drag and drop as well as text input. The participants were asked to use this tool to create and edit a rule that they can understand and agree with, while trying to increase the rule's accuracy.

3.1.2 Participants. We recruited 14 participants from a local university for the elicitation study. It was conducted over Zoom for 60 minutes. Participants were compensated $16 USD in local currency. They were genders 10 females and 4 males, ages 20 to 29 years old (M = 22.3, SD = 2.3) from 10 different discipline majors (3 in Computer Science and 3 in Business, and 1 for each of Chemistry, Medicine, Political Science).

3.1.3 Study procedure. The overall study consisted of two parts: i) editing stage where users were asked to create and edit a rule on their own to understand their interaction behavior using rules, ii) forward simulation stage to estimate their understanding of an AI model trained using their rules. In this section we focus only on the editing stage while the results from the forward simulation were collected and analysed in Section 6. The participants were shown 3 scenarios in random order: Income Prediction, House Price Prediction and Heart Disease Prediction. They built the rule and did the forward simulation for each of the scenarios.

During the editing stage, for each of the scenarios, they were given a guideline rule as background knowledge of the scenario, we used the same dataset and preparation procedure as described in Section 6. An accuracy indicator was available to them while they perform the editing. The participants were asked to think aloud to articulate their rationale behind their editing and to describe any difficulties they have encountered as well as their need of assistance.

3.2 Findings

All 14 participants successfully created and edited their rules for all 3 scenarios, we conducted a thematic analysis on participant behaviors and report key findings.

3.2.1 Preference for human-written rules over AI learned rules. Although the guideline rules achieved higher accuracy, most participants preferred to create and edit rules that reflected their own logic and domain knowledge. They valued explanations that "made sense" to them over rules that merely optimized prediction accuracy.

For instance, E5 reported that she *"actually just followed the guideline rule because I think it makes sense,"* but clarified that this was only because the rule's logic aligned with her reasoning: *"the size of the living area bigger than [a threshold] means higher value, and these also correspond with the grade. If it's older but in good condition, then it's high [price]."* In contrast, many others deliberately diverged from the guideline rules. E2 emphasized that logical consistency mattered more than accuracy, drawing on her own real-life knowledge: *"I started off with a high grade. And then I thought, what people want is a bigger living room… even if the living room is smaller, and the grade is smaller, more number of bedrooms can give high price."* She further explained, *"the more I focused on accuracy, like, the less I used logic. If I only focused on that, I might, like, come* *up with illogical choices."* Similarly, E3 rejected the guideline rules outright, noting, *"I'm following my thoughts. Because I think looking at the guideline rules, to me, it didn't really… make sense."*

Together, these accounts highlight that participants wanted the freedom to use rules that reflected their reasoning, even if these rules were less accurate. The ability to create and edit rules empowered them to maintain ownership of the decision logic, rather than feeling forced to adopt explanations they disagreed with.

3.2.2 Capped performance despite iterative edits. During the forward simulation stage, participants were allowed to revise their rules after seeing real instances and labels, and we logged their interactions for analysis. Most participants engaged in multiple rounds of editing, gradually revising their rules in search of improvement. For example, E2 described a step-by-step process of modifying thresholds and adding features until reaching a final model: *"I changed these 3 to… this from 4 to 3, and this from 3 to 2. And then I added this. The age, because I felt like high and low weren't doing, like, enough."* Similarly, E9 explained that he *"modified [the framework] based on the provided real labels because I noticed factors like house age and living area were not initially considered. I gradually revised the branches to improve it."* Despite such efforts, participants often struggled to achieve satisfactory accuracy. Iterative editing was described as tedious, and some resorted to drastic changes after several rounds of refinement. For instance, E8 abandoned her original rules after four rounds, stating that she *"wants to try out a different set of rule[s]."* Likewise, E7 and E14 restructured their rules late in the process when incremental edits failed to improve outcomes. These behaviors suggest that while iterative refinement is a natural strategy, it is insufficient without system support. Users require not only editing flexibility, but also suggestions and feedback tailored to their rules to sustain productive refinement.

3.2.3 Difficulty in determining threshold values and need for external advice. One of the most frequently requested forms of assistance was help with selecting appropriate threshold values for attributes. Participants often experimented with different cutoffs around an initial guess to see if accuracy improved. They sought guidance on ranges that were logical or grounded in domain knowledge, while still wanting to preserve the topology of their own rules. For instance, E1 remarked, *"I need to see [from the guideline], what range of values are logical,"* emphasizing the difficulty of choosing cutoffs without reference points. E7 similarly questioned what particular thresholds to be used, noting, *"I heard that you don't need to have a degree in order to get a high-paying job, but I don't know what's the cutoff for the age also."* Likewise, E11 expressed a desire for external assistance, suggesting:*"some help can tell you, for example, in the heart case, what is the normal resting [blood pressure] rate."*

These reflections underscore a recurring challenge: participants valued ownership of their rules but lacked confidence in setting precise numerical thresholds. This points to a clear design implication: the systems should provide threshold suggestions around the users' initial guess, while being contextualized by domain knowledge or data distributions, preserving the topology defined by users.

3.2.4 Need for reorganization suggestions while preserving some priors. Participants also recognized that their rules could benefit from topological improvements. Rather than discarding their work, they expressed interest in revisions: refining specific branches or adding detail while preserving the core of their original creation. For example, E3 reflected on his rule as *"I think it's incomplete."*, and would like to see where and how can improvements be made. Similarly, E5 struggled with how to make certain improvements, admitting, *"I'm not so sure about this [heart disease rule], I don't know exactly where to improve,"* and suggested that *"maybe the second resting blood pressure [needs improvement], but I don't know how."* E10 desired greater granularity on his rule, noting, *"I think my criteria could be more detailed, like maybe having more branches."*

These comments suggest that while participants valued the interpretability of their own topology, they lacked confidence in knowing where or how to revise them. This highlights the need for system support that can propose structural refinements, while maintaining alignment with users' original rule.

3.3 Design Guidelines

Based on these findings, we propose design guidelines for **Editable XAI** to facilitate human-AI alignment:

1) **Contextualize explanations in a writable format** such as rules so that users can contribute to adjusting the explanations rather than just read them.
2) **Support user-written rules** by explaining in terms of rules established in the domain, or defined by users as they need, instead of purely learned from data.
3) **Provide automated enhancements of user-written rules** to alleviate knowledge deficits of users, and augment their reasoning. The system can recommend adjustments to rule thresholds (parameters), or even reorganize the rule topology (decision tree structure) to significantly improve the explanation accuracy.
4) **Constrain enhancements** to preserve alignment with user-driven rules and limit explanation complexity.

4 TECHNICAL APPROACH

To satisfy the design guidelines, we introduce **CoExplain** for collaborative, editable explainable AI (XAI). We model predictions using neural networks to leverage its representational power as universal function approximators [109], and gradient descent methods for model training. We explain the model using decision trees to leverage its intuitive representation of semantic relationships [90]. As illustrated in Fig. 2, CoExplain enables users to *read* explanations through **distillation** (Section 4.1), *write* explanations through **parsing** (Section 4.2), and the AI can *enhance* explanations through **training** with backpropagation and regularization (Section 4.3).

4.1 Distillation: Neural Network Model to Decision Tree Rule Explanation

Model agnostic explanations, such as LIME [105], are commonly used to explain black box AI models like neural networks. These can be done for local explanations with instances near a target, or global explanations for the overall model decisions. Generally, this approach is *model distillation*, most commonly used to condense large models (e.g., student-teacher models [47, 124]), but can also be used for explanation since distilled models are simpler. For this work, we focus on explaining neural networks with a global decision tree using distillation [15, 28, 71, 99, 110].

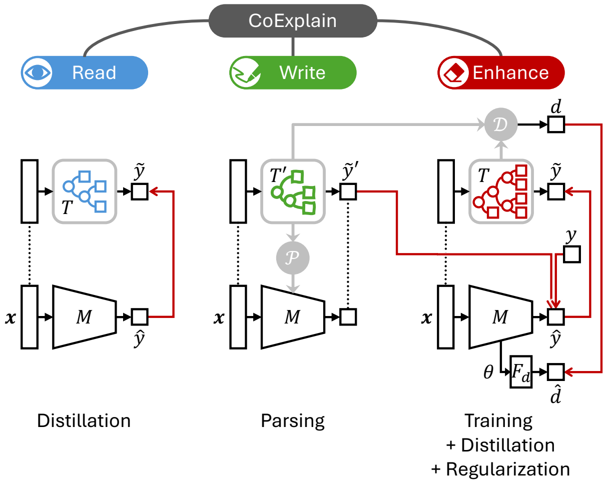


Figure 2: Overview of CoExplain's three interaction modes and their underlying mechanisms. Read: explanations are distilled from the neural network M into a decision tree T, using the input x and network prediction $\hat{y}$ to make sure tree prediction $\tilde{y}$ aligns with M. Write: user-authored rules T' are transformed into a neural network M through a parser $\mathcal{P}$. Enhance: user collaborates with AI edits on the thresholds and topology. AI refines the rules through training with regularization, aligning predictions $\hat{y}$ with both user-defined rules' prediction $\tilde{y}'$ and data-driven adjustments with y, the topology of the explanation T is aligned with T' using a proxy model F_d mapping the network parameters θ to their Tree Edit Distance d calculated by $\mathcal{D}$.

Consider the AI model (neural network) to be explained M, and explainer XAI model (decision tree) T. Predictor model M is trained on instances $\boldsymbol{x}$ with ground truth label y, and predicts label $\hat{y}$, i.e., $\hat{y} = M(\boldsymbol{x})$. Instead of training explainer model T on ground truth labels, we train it on the prediction label $\hat{y}$ to predict explanation label $\tilde{y}$, i.e., $\tilde{y} = T(\boldsymbol{x})$. We performed hyperparameter tuning of the tree depth to obtain the most faithful explainer model T to predictor model M. The neural network M is trained via backpropagation, while the decision tree T is trained via CART [89].

4.2 Parsing: Decision Tree Rule Explanation to Neural Network Model

While the distillation focuses on explaining a pre-existing neural network, parsing starts with a pre-existing decision tree written by a user. Moreover, while distillation is model agnostic and approximate, parsing is heuristic and can *exactly convert* a tree into a neural network, exploiting the property that both are graph structures.

We build on prior work in tree-to-network initialization [60] and neural-symbolic alignment [82] to parse a user-defined decision tree into a topologically-equivalent neural network. Note that while both trees and neural networks have graph structures, their aspects do not represent the same concepts. Table 1 compares key aspects between these models. Next, we describe how to parse aspects of the decision tree (decision nodes, trace conjoined branches, disjoint traces), and how to ensure extensibility of the tree-based neural network. See Fig. 3 for an illustrated example of parsing.

4.2.1 Decision Nodes and Consequent Branches. For a binary decision tree, each decision node results in a True (T) or False (F) outcome. We model these in the first layer of the neural network, where each outcome is associated with a neuron, and each decision node has a pair of neurons. So, k decision nodes results in a first NN layer with $2k$ neurons.

From Fig. 3, consider the blue decision node with test $x_1 \geq \tau_1$. The test is True when $x_1 - \tau_1 \geq 0$ and False when $-(x_1 - \tau_1) > 0$. These results can be represented in neurons as $a_{11} = \mathbb{I}(x_1 - \tau_1) = \mathbb{I}((-1, +1) \cdot (\tau_1, x_1))$ and $a_{12} = \mathbb{I}(-(x_1 - \tau_1)) = \mathbb{I}((+1, -1) \cdot (\tau_1, x_1))$, respectively; where $\mathbb{I}(P)$ is the indicator function which is 1 when P is true and 0 otherwise. Similarly, the green decision node $x_2 \geq \tau_2$ is represented by neurons $a_{13} = \mathbb{I}((-1, +1) \cdot (\tau_1, x_1))$ and $a_{14} = \mathbb{I}((+1, -1) \cdot (\tau_1, x_1))$. Represented in parallel, in NN layer 1 and subsituting $\mathbb{I}$ with a differentiable smooth sigmoid activation function (σ), both decision nodes are:

$$\boldsymbol{a}_1 = \sigma\left(\boldsymbol{W}_1^{\top}\boldsymbol{x} + \boldsymbol{b}_1\right),$$

$$\begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{pmatrix} = \sigma\left(\begin{pmatrix} +1 & 0 \\ -1 & 0 \\ 0 & +1 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -\tau_1 \\ +\tau_1 \\ -\tau_2 \\ +\tau_2 \end{pmatrix}\right), \tag{1}$$

with input $\boldsymbol{x}$, weights $\boldsymbol{W}_1$, and biases $\boldsymbol{b}_1$.

4.2.2 Conjunction along a Trace. Although represented together in $\boldsymbol{a}_1$ (Eq. 5), each branch of each decision node is treated separately. To represent traversing down the tree through multiple decision nodes, we need to encode logical *conjunction* (AND $\wedge$). We do so with Łukasiewicz conjunction to handle continuous values:

$$P \wedge Q = \max(0, p + q - 1), \tag{2}$$

where $p = 1$ when P is True and 0 otherwise, and same for q with Q. We encode this operation in the second NN layer by representing $p + q - 1$ in a computational graph with ReLU activation function since $\mathrm{ReLU}(x) = \max(0, x)$. As an example, the conjunction $(x_1 \geq \tau_1) \wedge (x_2 \geq \tau_2)$ in the second depth of the decision tree in Fig. 3 is represented by neuron $a_{23} = \mathrm{ReLU}(a_{12} + a_{13} - 1)$. All decision branches are represented in the NN layer 2 as:

$$\boldsymbol{a}_2 = \mathrm{ReLU}\left(\boldsymbol{W}_2^{\top}\boldsymbol{a}_1 + \boldsymbol{b}_2\right),$$

$$\begin{pmatrix} a_{21} \\ a_{23} \\ a_{24} \end{pmatrix} = \mathrm{ReLU}\left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix}\right), \tag{3}$$

with input $\boldsymbol{a}_1$, weights $\boldsymbol{W}_2$, and biases $\boldsymbol{b}_2$.

For DT traces that terminate at the first or earlier tree depths, the NN propagations *pass through* the layer without modification (i.e., weights = 1 and biases = 0). For example, decision branch $x_1 \geq \tau_1 \models \hat{y}$ has the NN propagation $\hat{y}_{\mathrm{I}} = 1 \cdot a_{11} + 0$.

4.2.3 Disjunction of Traces with Same Leaf Node Predictions. We represent multiclass classification using one-hot encoding, where each neuron in the NN output layer is a separate class. When multiple traces in the decision tree share the same predictions in their leaf nodes, this indicates alternative antecedents for the same consequent, which is a logical *disjunction* (OR $\vee$). We encode this

Table 1: Comparison of graph structures between a decision tree and its topologically equivalent neural network.

Aspect	Decision Tree (DT)	Neural Network (NN)
Node	**Decision Node** representing a test on a single attribute against a threshold. **Leaf Node:** Represents the predicted label from the preceding branch outcome (e.g., $x_1 > 5$).	**Neuron** representing a unit that receives weighted inputs and activates an output.
Edge	**Branch** representing the binary prediction outcome, connecting to the next decision test.	**Connection** representing the weight of the prior input (neuron).
Path	**Trace** through multiple decision nodes represents the conjunction (AND) of tests. If multiple traces have the same predicted label in their leaf nodes, then the traces can be combined as a disjunction (OR).	**Propagation** of multiple computed operations from input to output.
Level	**Depth** of a node corresponds to the number of attribute tests required to reach it.	**Layer:** For the parsed NN, the first layer represents straight (linear) decision boundaries, and subsequent layers are weighted, piecewise combinations of these boundaries. General NNs learn nonlinear decision boundaries or more separable feature spaces.
Computation	**Symbolic:** Each node performs a symbolic test (e.g., $x_2 > 8$).	**Continuous:** Each neuron performs a nonlinear operation (weighted sum and activation function).

numerically with Łukasiewicz disjunction:

$$P_1 \vee P_2 \vee \cdots = \min(1, \sum_{t \in T_c} p_t), \tag{4}$$

where $t \in T_c$ is the tth trace that predicts class c, and $p_t = 1$ when P is True and 0 otherwise. We encode this in the output layer by summing relevant activations from the penultimate layer with Clipped ReLU (cReLU) activation function: $\mathrm{cReLU}(x) = \min(1, x)$ for clip at 1 and $0 < x \leq 1$. All decision traces are represented in the output layer as:

$$\begin{gathered} \hat{\boldsymbol{y}} = \mathrm{cReLU}\left(\boldsymbol{W}_y^\top \boldsymbol{a}_2 + \boldsymbol{b}_y\right), \\ \begin{pmatrix} \hat{y}_\mathrm{I} \\ \hat{y}_\mathrm{II} \end{pmatrix} = \mathrm{cReLU}\left(\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} a_{21} \\ a_{23} \\ a_{24} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right), \end{gathered} \tag{5}$$

with input $\boldsymbol{a}_2$, weights $\boldsymbol{W}_y$, and biases $\boldsymbol{b}_y$.

4.2.4 Model Capacity and Zero Padding. The aforementioned methods describe how to set parameter values (weights and biases) in the neural network, but do not specify the model capacity needed to represent the parsed tree or future extensions. At minimum, i) the number of neurons in each NN layer is twice the number of DT decision nodes, and ii) the number of NN layers is determined by the depth of the DT. The number of neurons in NN output layer is the number of prediction classes.

However, to support extensibility whereby the model can be retrained to be more expressive and accurate, we need to support more relationships and decision boundaries. We do so by *zero padding* with more neurons per layer (weights = 0, bias = 0), and more layers with *pass through*. Section 4.3 describes how training to update these additional parameters can optimize the model further.

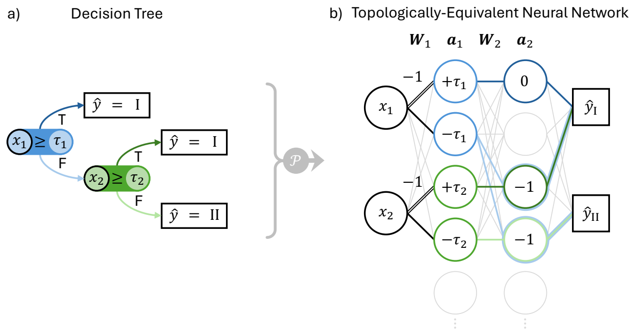


Figure 3: Transforming a decision tree into a topologically equivalent neural network through a parser $\mathcal{P}$. (a) A decision tree with two internal nodes, each testing a feature threshold. (b) The corresponding neural network, where each tree node is mapped to a pair of neurons in the first hidden layer with biases $\pm\tau_i$ to encode the threshold test. Subsequent layers mirror the decision paths of the tree: connections are preserved with weights of 1, while neuron biases are set according to the logical relationships. Colored connections trace decision paths, and the output layer aggregates signals to reproduce the tree's leaf predictions.

4.3 Training: Optimizing Model for Performance and Alignment

User-created decision trees provide an interpretable starting point, but their predictive performance is often limited by the greedy[1], local optimization used during construction. To manage user sense of control and moderate edits, CoExplain supports conservative and disruptive update methods: Threshold Enhancement and Topology Enhancement. These change the underlying neural network model

[1] Due to CART-based learning with information gain.

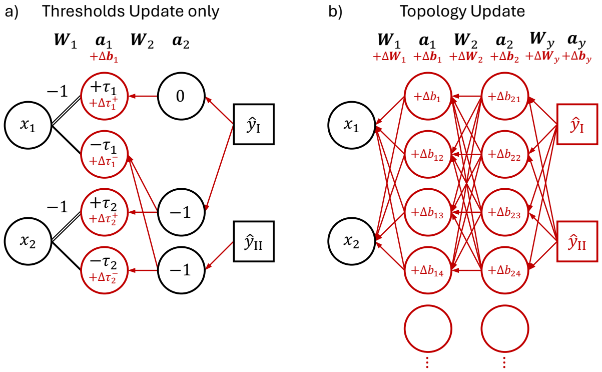


Figure 4: Two types of enhancement from CoExplain, a) Threshold Update, only the threshold is updated, while keeping the topology still. b) Topology Update, both the thresholds and connections are trained, we utilize additional neurons and connections to support decision tree extension. Backpropagation and parameter update are marked as red.

slightly or by a lot and explain with a new decision tree as updated. Threshold Enhancement will retain the same decision tree structure (topology), but may change the threshold value (e.g., $x_1 \geq 5$ may become $x_1 \geq 5.5$). Topology Enhancement will learn complex, nonlinear, multivariate relationships, resulting in a change in topology, such as adding and rearranging decision nodes, and using additional attributes. To limit the changes, constraints can be applied. Next, we describe how CoExplain enhance through model training, distills explanations for validation, and aligns to constraints via regularization (see Fig. 2 Enhancement for architecture).

4.3.1 Training. We retrain the neural network in the standard approach using supervised learning and gradient descent via backpropagation on a training dataset using cross-entropy loss between the model’s prediction $\hat{y}$ and ground truth y, i.e., $\mathcal{L}_\text{data}(\hat{y}, y)$. Enhancements for thresholds and topology are handled differently.

Threshold Enhancement: To enhance only the DT thresholds, we freeze all parameters except for the bias terms in the NN layer 1 that contain the DT thresholds (see Fig. 4a). For example, $b_{11} = +\tau_1$ can be updated with $\Delta\tau_1$. Moreover, a user may wish to lock in an edited threshold. To preserve this user-defined threshold, and ensure local consistency, we freeze the bias terms of its two corresponding neurons in the first layer. For example, locking the node $x_1 \geq \tau_1$ in Fig. 3a will freeze the bias terms in the top two neurons of layer $\boldsymbol{a}_1$ in Fig.4a.

Topology Enhancement: To further enhance the DT toplogy, we unfreeze all layers and parameters and allow full weight and bias updates (see Fig. 4b). This includes updating the zero-padded neurons. Moreover, instead of letting CoExplain fully alter the full decision tree, the user may want only some rules to be enhanced and others restricted. To preserve the restricted rules, we selectively freeze those rules as we freeze threshold neuron biases. However, this may be overly restrictive, leading to low accuracy. Instead, to discourage the AI from changing restricted rules, we increase their weight in the Tree Edit Distance regularization, so that changes to restricted rules have a higher penalty than changes to other rules.

4.3.2 Distillation. To generate the decision tree explanation T of the retrained model M, we use the same distillation method described in Section 4.1. However, we do this iteratively at each training batch as an intermediate check on the performance and topology of the explainer model.

4.3.3 Regularization. Although standard supervised learning focuses on model performance, CoExplain also supports alignment to user-defined rules. This is accomplished by referring to the user’s written decision tree T', and using self-supervised learning to ensure that the distilled tree explanation T has similar behavior ($\tilde{y} \approx \tilde{y}'$) and topology design. Fig. 2 We regularize behavior with cross-entropy loss between the DT explainer’s prediction $\tilde{y}$ and user DT prediction $\tilde{y}$, i.e., $\mathcal{L}_\text{behavior}(\tilde{y}, \tilde{y}')$.

To determine the difference in topology between T and T', we calculate the Tree Edit Distance (TED) d which counts the number of edit operations (insert, delete, update) needed to change from one tree to the other. Specifically, we used the ZSS algorithm [128] implemented in Python. Since this distillation to obtain T from M is external to the neural network, we cannot use T directly for training with gradient descent. Instead, we trained a proxy model F_d that uses the parameters θ in M as input to predict the TED distance $\hat{d}$. Hence, M is a multitask model to predict $(\hat{y}, \hat{d})$ and it updates its parameters to optimize for both tasks. We regularize topology with MSE loss $\mathcal{L}_\text{topology}(\hat{d}, d)$.

In summary, to align the model to training data and user-defined rules we have multiple loss terms:

$$\mathcal{L} = \mathcal{L}_\text{data}(\hat{y}, y) + \lambda_\text{b}\mathcal{L}_\text{behavior}(\tilde{y}, \tilde{y}') + \lambda_\text{t}\mathcal{L}_\text{topology}(\hat{d}, d) \quad (6)$$

where λ_b and λ_t are hyperparameters that we let users control as similarity settings.

4.4 Implementation Details

We implemented the decision tree components using `scikit-learn`, using CART [16] as tree learner. The neural network components were implemented in `PyTorch` and optimized with Adam (learning rate 1×10^{-2}). All experiments were conducted on a workstation equipped with an NVIDIA H100 PCIe GPU (81,559 MiB memory) and an AMD EPYC 9654 96-core processor.

5 INTERFACE DESIGN

Fig. 5 shows the user interface for the CoExplain rule explanation editor. The interface has a key component for reading and editing rules, complemented by panels that display dataset attributes, model performance, and AI enhancement.

5.1 Data Attributes

The attribute list (Fig. 5a) lists available dataset features. Both user and the AI use attributes from this list when editing.

5.2 Explanation Rules Canvas

The core component is the rules canvas (Fig. 5b), where users read and edit the AI explanation as rules represented in a decision tree. Each rule is visualized as a graphical flowchart: nodes denote decision antecedents, while edges represent true (green, top) or false (red, bottom) consequents. We considered presenting the rules in

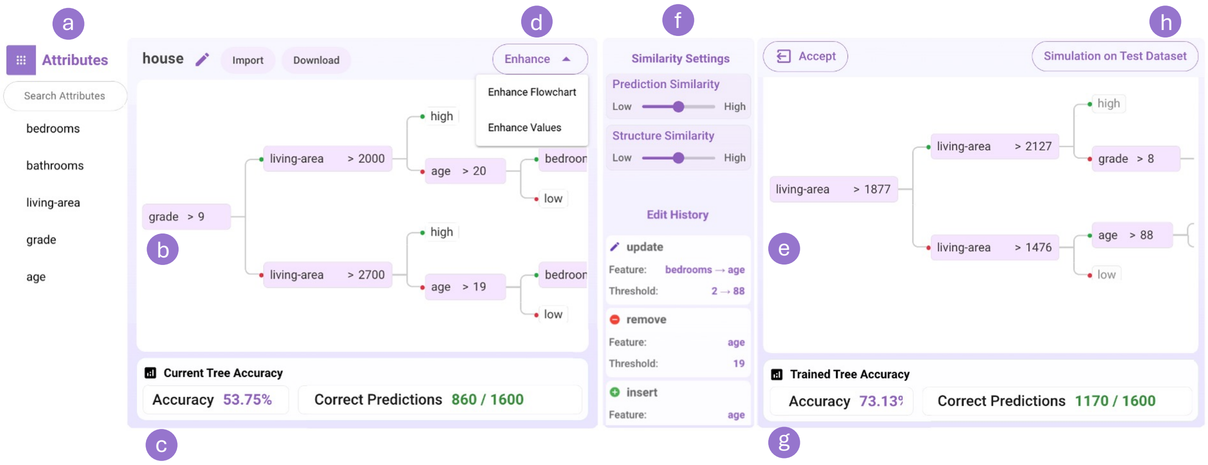


Figure 5: Interface of CoExplain. a) Data attributes. b) User's explanation rules canvas. c) User rule performance metrics. d) Enhancement actions. e) AI-enhanced explanation rules canvas. f) Enhancement constraints and edit history. g) Enhanced rule performance metrics. h) Simulation on test dataset.

programming syntax, a format often used in end-user development. However, pilot users found the flowchart representation more intuitive. We abstract away the underlying neural network implementation. In the subsequent user study, participants did not request such details; the one technical participant who inquired was surprised to learn about the relationship between his rules and the neural network, but expressed no need for further inspection.

5.3 Model Performance Metrics

Performance metrics are displayed below each tree (Fig. 5c and g). These include accuracy on the training set and the number of correctly predicted cases, allowing users to compare their current tree with the AI-enhanced alternative.

5.4 Enhancement Aids

To support the user with CoExplain enhancements, the interface displays a parallel AI-edited rule canvas (Fig. 5e). Users can choose to accept the entire AI editing or selectively incorporate parts that they agree with.

5.4.1 Enhancement Actions. Enhancement actions are accessed via a main button with a dropdown (Fig. 5d). Two sub-actions are available: 1) Threshold Parameters (Value) Enhancement, and 2) Topology (Flowchart) Enhancement.

5.4.2 Enhancement Constraints. Users can regulate how the AI proposes changes via two similarity sliders (Fig. 5f). *Prediction Similarity* controls how closely the enhanced tree should match the current tree's predictions on the test dataset. *Structure Similarity* controls how similar the enhanced topology should remain to the user's current tree. Together, these values determine the hyperparameters of the regularization terms in Eq. 6.

5.4.3 Edit Operations History. The interface records a complete edit history performed by the AI (Fig. 5f), showing insert, update, and remove operations. This allows users to track how the enhanced tree diverges from the original one.

5.5 Simulation on Test Dataset

Finally, the simulation button (Fig. 5h) lets users test their tree on an explicit test dataset.

6 EVALUATION

We conducted a user study to evaluate the degree of alignment and interpretability of our proposed editable explanation compared with Read-only explanations and manual editing variations. Our method and its variations were tested on three scenarios with real-world datasets, qualitative and quantitative analysis were conducted on the collected data.

6.1 Method

We describe the method of our experiment, including the experiment design, application tasks, background knowledge context, experiment procedure, and recruited participants.

6.1.1 Experiment Design. We primarily compare three XAI Types—variants of the explainable AI interface with different explanation capabilities:

- **Read-only** shows a global explanation of the AI model as a decision tree, which the user can only **Read** and not modify, examples of Read-only explanation rules can be found in Appendix A.5.
- **Editable** represents a non-collaborative setting, where participants **Write** and edit their rules without AI assistance. They are provided with an accuracy indicator of their rules, consistent with the one used in Section 3.
- **CoExplain** allows the participant to collaborate with the AI to co-edit the explanatory rules of the model. In addition to **Read** and **Write** facilities, the participant can ask CoExplain to **Enhance** the explained rules to increase performance on a training dataset.

The experiment design comprises **XAI Type** as the primary independent variable (IV), decision with or without XAI (**w/ XAI**) as a secondary IV, and application **Dataset** as a random variable (RV). We presented each XAI Type *between-subjects* to avoid the learning effect from prior exposure to different explanation features, and misattributing opinions to wrong systems. To measure participant understanding, we ask participants to estimate the AI decision (measures described in Section 6.1.2, procedure described in Section 6.1.5) without then with XAI, *sequentially* to avoid the learning effect of being shown more information first. For generality, we also varied the application task for which the participant uses the XAI UI. This was presented *within-subjects* in random order (counterbalanced for three applications) to allow repeated measures of different applications per participant to reduce individual variance.

6.1.2 Measures. We measured objective metrics of user understanding (U), AI performance (P), AI alignment (A), engagement (E), and subjective perceived ratings (Q):

U1) **User-AI Faithfulness** of whether the participants label was the same to the AI prediction label (1 if same, 0 if different). This indicates if they can correctly anticipate the AI's decision, as user understanding of the AI.

P2) **AI Accuracy (in Pre-trained Distribution)** of how well the final AI model (if edited) performs on a test set from the Pre-trained distribution. This indicates if the AI performs well in its original training domain.

A3) **AI Accuracy (in Guideline Distribution)** of how well the final AI model (if edited) performs on a test set the target Guideline distribution. This indicates if the AI is aligned to target domain as desired by the user.

A4) **Distance between XAI and Guideline Rules** of how similar are the Guideline Rules and the distilled Decision Tree explanation of the final AI model. We computed distance using the Tree Edit Distance (TED) [128]. This indicates if the XAI is aligned to target guidelines as desired by the user.

E5) # **Edit Operations** indicating how many changes the participant made to the XAI rules.

E6) # **Edit Iterations** indicating how many iterations the participant went through before finalizing the AI.

E7) **Editing Time** indicating how long all editing took. We log transformed Editing Time due to the long tail of slower participants, to make the response better satisfy the normality assumption for statistical analysis.

R7) **Perceived Ratings** along a 7-pt Likert scale (−3 Strongly Disagree to +3 Strongly Agree) on the participant perceptions regarding:

- Understanding of AI Rules (decision tree)
- Understanding of AI Decisions (predictions)
- Ease of Use to Edit XAI (Editable XAI and CoExplain)
- Ease of Use of CoExplain Enhancement
- Alignment of CoExplain Enhancement

6.1.3 Application Tasks. We selected three application tasks and their dataset for evaluation: *Adult Income* [38], *House Price Prediction* from the "House Sales in King County, USA" dataset [50], and *Heart Disease* from the UCI Machine Learning Repository [63]. The *Adult Income* task has been widely used as a testbed for human-AI collaborative decision-making [43, 51, 91], while the other two datasets follow prior work by Bo et al. [13]. The datasets vary in domain knowledge requirements: the *Adult Income* and *House Price* tasks rely on general knowledge and are therefore more accessible to lay participants, while the *Heart Disease* task requires professional training beyond commonsense. To balance complexity and usability for the participants [13, 43, 91], we selected a small set of features: 5 for *Adult Income* (Age, Education, Marital status, Investment gain, Weekly hours), 5 for *House Price* (Bedrooms, Bathrooms, Living area, Grade, Age), and 4 for *Heart Disease* (Age, Resting blood pressure, Cholesterol, Max heart rate). We partition each application task's data into two distributions: a guideline distribution used to provide background knowledge to participants, and a pre-trained distribution that participants haven't been trained on while it's used to train the AI. This creates a controlled misalignment between the human and AI, as discussed in Appendix A.1.

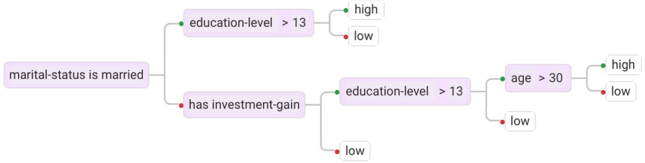


Figure 6: Guideline rule for *Adult Income*, green dot indicates true branch, red indicates false branch.

6.1.4 Background Knowledge: Guideline Rules. For each application task, we provide participants with a guideline rule as the starting point for constructing their rules. This guideline rule serves as a controlled background knowledge and is based on the guideline distribution. The rules are obtained by training a neural network on the guideline distribution for five iterations to ensure fairness with the Read-only, which adopt the same setting. Afterward, the distillation method described in Section 4.1 is applied to extract decision-tree rules from the trained network. Fig 6 illustrates a guideline rule from the Adult Income Prediction Task. The guideline rules for the other tasks are provided in Appendix A.7.

6.1.5 Procedure. Each participant went through the following:

1) Introduction (5 min). Participants were briefed on the research background, motivation, and study protocol. Informed consent was obtained to record their interactions and screen-sharing for analysis, the University ethics review board approved human-subjects research and they approved this project.
2) Tutorial and Screening (5 min). Participants received an introduction to decision rules and the AI system functions, followed by a short screening task to ensure comprehension.
3) Main Session (50 min). Participants completed three scenarios in random order. For each scenario:
   a) Introduction to the attribute definitions and guideline rules for the scenario.
   b) Forward simulation across 20 case trials, consisting of:
      i) Rule editing on the explanation (Editable, CoExplain only).
      ii) Estimation of the AI's prediction without explanation.
      iii) Estimation of the AI's prediction with explanation.
      iv) Observation of the AI's prediction and ground-truth label.
   c) Perceived ratings and a short interview.

4) Demographics, Debrief, and Compensation. Participants completed a demographics questionnaire, were debriefed of their performance, and received compensation.

6.1.6 Participants. Via a university mailing list, we recruited 43 participants from a local university (16 males, 27 females), aged between 19 and 30 years old (M = 22.3, SD = 2.5), with diverse disciplinary backgrounds from 18 different majors (e.g., Psychology, Business, Political Science, Computing, Medicine). Participants were randomly assigned to one of three variants. We recruited 15 participants per condition, due to no-shows the final distribution was 14, 14, and 15 participants for **Editable** (E1–E14), **CoExplain** (C15–C28) and **Read-only** (R29–R43). This minor imbalance does not substantially affect the comparability of conditions. The study was conducted through an online audio call, with participants consent, we recorded the meeting and their screen-sharing during the study. The study lasted for around 1 hour and participants were compensated $16 USD in local currency.

6.2 Qualitative Findings

To investigate how users engage with **CoExplain** and benefit from editable explanations, we conducted a qualitative analysis on their interactions and post-trial interview. We organize our findings in terms of the edit operations of writing to i) align the AI to the user's knowledge, ii) actively engage user understanding of the AI decisions, and AI-enhancement to update iii) basic thresholds and iv) complex topology.

6.2.1 Editing Aligns AI Reasoning with Users' Knowledge. Participants found the decision rules **clear and easy to follow**, which made them accessible. R38 described the rules as *"easily understandable"* with *"clear indicators of what orders to follow according to quantifiable metrics"*. C23 found rule creation approachable, stating, *"If you satisfy the first [condition], then look at the next, and so on."*

Participants considered the AI's outputs as closely **mirroring their own reasoning**. E12 explained her reasoning process on house price prediction, *"I followed the logic of house age first (older = lower price), then grade, then living area, so the AI's behavior matches my thoughts."* E14 similarly observed, *"AI's predictions mirrored my thought process."* These comments reveal that editing allowed participants to directly reflect their reasoning in the AI's behavior.

Editing also enabled participants to align the AI with their expectations, which **increased predictability** and **perceived reliability**. C25 noted, *"AI's prediction and my prediction are in line. I can trust AI's prediction."* E13 added, *"When I adjusted the rules, the AI's predictions changed to match. It's like the AI 'thinks' how I do."* These reflections show that editing not only align AI with user reasoning but also fostered user confidence in the AI's reasoning.

Some participants went beyond the provided guidelines, integrating their own domain knowledge. For instance, C21, a Pharmacy student, noted during the heart disease prediction task, *"I think the AI is generally following me in terms of my beliefs, as well as the values that I'm using. Like, it's quite congruent with what I'm thinking."* This demonstrates that rule editing can empower participants to project personal expertise into the AI. However, alignment is not always beneficial: when users lack domain knowledge or misinterpret rules, the AI can reinforce incorrect reasoning.

Overall, these findings indicate that editing supports active engagement, allowing participants to shape the AI's reasoning in ways that improve understanding, predictability, and trust.

6.2.2 Editing Improves User Understanding through Active Engagement. With **Read-only** Explanation, some participants struggled to make sense of the AI's reasoning because they could not adjust the rules. As R42 noted, *"some portions don't really make sense, like how having more bedrooms at the end made it low when logically it should be high."* Without the ability to intervene, participants often resorted to guessing patterns, as R43 described, *"There were some attributes that were more repeatedly affecting decisions than other so [I] started focusing on them."*

Allowing users to directly modify rules with **Editable** explanation substantially improved their understanding. Many emphasized that comprehension came naturally because they wrote the rules themselves. E10 explained, *"The judgment process was written by myself, so I know which indicators will lead to which results."* As E13 put it, *"I know the rule well because I created it."* These reflections show how the act of editing turned explanation into **active engagement**, strengthening users' understanding about the AI.

CoExplain went further by combining user edits with AI enhancements, enabling participants to see beyond their own assumptions. C21 described a shift: *"At first, I didn't understand why the AI didn't prioritize age… now I see age isn't a strong factor, so it makes sense."* C15 echoed this process of adaptation, acquiescing to the AI's logic during the house price prediction task: *"The AI kept rejecting [the age attribute], so maybe I should just follow the AI and see."* Through such iterations, participants came to understand not only their own rules but also the AI's reasoning. Yet, enhancements sometimes created confusion. C25 recalled, *"During the enhancement, the AI suggested that I swap the age and the cholesterol level, it was not my original order, I think I am a bit confused."*

Together, these accounts show that editing, whether with self-authored rules or collaborative enhancement, helped participants grasp the AI's reasoning more effectively than passive observation.

6.2.3 Appreciation for the Advice of Conservative Updates on XAI Rule Parameters. Participants valued CoExplain's enhancement to thresholds when those adjustments aligned with their knowledge. When working on the medical task, C21 welcomed help with unfamiliar medical ranges, *"Because I'm not familiar with the numbers. For example, the max heart rate, so I'll accept that one."* Participants regarded these edits as "light touches" that calibrated rather than replaced their intentions, as C23 put it, *"the values don't differ by a lot. I think the values are actually quite close to mine".*

When threshold changes aligned with plausible expectations, participants found them credible and useful. C19 grounded this reaction in housing knowledge, *"A grade 4 house is terrible, so it sells low; a 98-year-old house is a fire risk—this aligns with what I'd expect."* In health scenarios, concrete anchors made threshold choices feel self-evident: C17 noted, *"If you see someone with a 300 cholesterol level, there's something wrong."*

Participants also appreciated when suggested thresholds added clarity to their distinctions, as C25 reflected after getting his threshold rounded for easier interpretability (e.g. 38 changed to 40 for age) on the Income Prediction task, *"straightforward with clear cut-offs."*

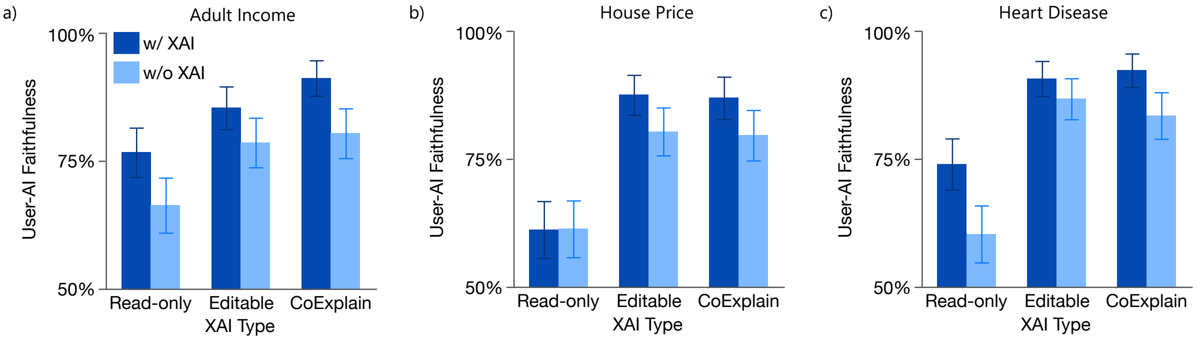


Figure 7: Results from forward simulation trials for users to estimate the AI system's prediction with and without viewing explanations on a) Adult Income, b) House Price, and c) Heart Disease.

Not all changes, however, were viewed positively. A few participants felt some adjustments were either too minor to matter or misaligned with their expectations, for example, C25 noted on heart disease prediction task, *"the value change for resting blood pressure is very small... and for the cholesterol level the change is very significant, but I don't think it is correct."* Such reactions suggest that conservative updates were most effective when they align with participants' domain understanding.

Across these cases, threshold enhancements were accepted when they fit participants' knowledge, clarified their boundaries, and addressed uncertainty without undermining ownership.

6.2.4 Topological Enhancements Enables Users to Refine Rules under their Control. Participants frequently described topological enhancement as making rules clearer and easier to work with, while still retaining control over key elements. C25 praised the **legibility** and **conciseness** of the enhanced rules which makes it easier to understand, *"more simplified and easy to understand at a glance."*

C16 framed reorganization as providing a second opinion for verification, *"it gives you another flowchart for cross-reference, maybe some branches are redundant, and you could improve."* Participants used these structural views to streamline logic and remove clutter.

At the same time, participants actively **moderated** how much restructuring to allow. C19 reported intentionally widening leeway, *"I lowered the [structure] similarity to the lowest because I don't need the AI to stick to my structure."* C15 explained using constraints to preserve a belief, *"I wanted to make the AI still follow the attribute, both similarities need to be high enough."* Participants also articulated trade-offs. C26 reflected on domain-contingent deference, *"I trusted my own judgment in familiar areas (like house prices),"* while still appreciating support where knowledge was thinner.

Resistance to structural edits surfaced when participants felt important features were dropped. C17 objected to removals, *"Sometimes it didn't really enhance it because it dropped the bathroom and bedroom attributes."* C24 preferred complexity and accepted wholesale restructuring as, *"I didn't insist on my own structure because I think the AI will just give me a better one. It's very complex, and my own one is very simple."* Together these accounts show participants selectively embracing simplification and reorganization while exercising granular control over what should remain intact.

6.2.5 Take-away. Overall, participants used editing to align the system's reasoning with their own and to understand how and why it reasoned differently when it did. They welcomed threshold and structural enhancements that reflected their knowledge, while actively moderating changes. Editing and enhancements together supported **bi-directional alignment**: users shaped the AI with their rules and learnt how it refined and extended their reasoning.

6.3 Quantitative Results

We evaluated how Editable and CoExplain explanations affect user understanding, alignment, and engagement compared with Read-only explanations. We consider results with $p \leq .002$ as statistically significant to account for up to 25 multiple comparisons up and avoid Type I error. Our analyses address four questions: i) do Editable and CoExplain improve user-AI faithfulness, ii) how do they influence user-AI alignment and model performance, iii) how do they shape user engagement, and iv) how are these explanations perceived by participants.

6.3.1 Statistical Analysis. For statistical analysis, binary measures P1, A2, and A3 were analyzed using a Generalized Linear Mixed Model (GLMM) with a binomial distribution and logit link. Fixed effects included *XAI Type*, *Dataset*, whether the trial was *with or without XAI*, and their interactions, with Participant ID as a random effect. For all other measures, we fit Linear Mixed-Effects Models with *XAI Type*, *Dataset*, and their interaction as fixed effects, and Participant ID as a random effect, using restricted maximum likelihood (REML) estimation. We examined statistical controls with demographic and background variables (gender, major, age, domain familiarity, and prior AI experience) as covariates, but found no exhibited significant effects. We thus omitted them from our statsitical models. We report significant differences (p < .001), and their effect size using Cohen's d [30] from the baseline Read-only condition.

6.3.2 Editable Explanations Improve User-AI Faithfulness. We found significant main effects of *XAI Type* ($p < .0001$), *w/ XAI* ($p < .0001$), and *Dataset* ($p = 0.0003$). Post-hoc contrast tests showed that both Editable ($p < .0001$) and CoExplain ($p < .0001$) significantly improved user-AI faithfulness compared to Read-only, while not differing significantly from each other.

Adult Income. As shown in Fig. 7a, Read-only participants performed the worst (M = 66.6% without XAI; 77.1% with XAI). Editable performed better overall (M = 80.6% and 87.5%), showing large advantages over Read-only in both settings ($d = 0.94$ and $d = 0.87$). CoExplain achieved similarly strong results (M = 81.4% and 91.9%),

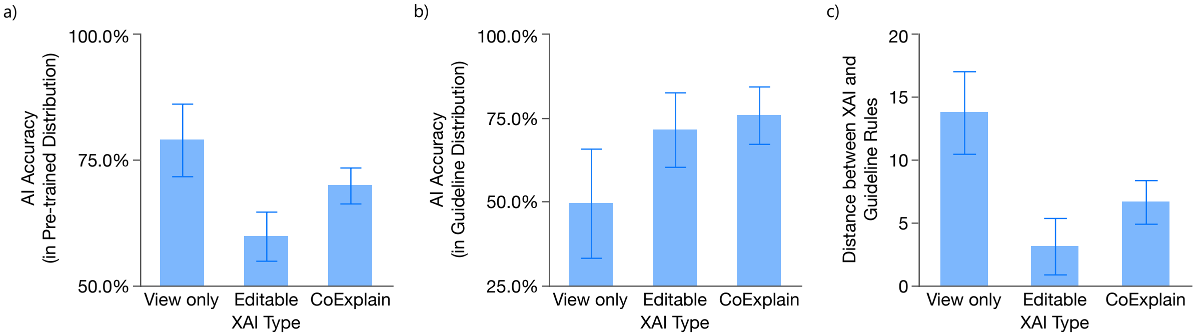


Figure 8: Results measuring AI performance on pre-trained and guideline distribution and alignment with guideline rules across different XAI types: a) AI accuracy on the pre-trained distribution, b) AI accuracy in the guideline distribution, and c) distance between XAI explanations and guideline rules.

again with large differences relative to Read-only (d = 0.98 and d = 1.32). Both Editable and CoExplain thus fostered substantially better user understanding than Read-only.

House Price. Editable participants achieved high faithfulness (M = 82.5% and 89.6%), with large effects relative to Read-only (d = 1.37 and d = 2.01). CoExplain showed similarly strong performance (M = 80.6% and 87.6%), again yielding large differences from Read-only (d = 1.21 and d = 1.80). By contrast, Read-only participants showed minor improvement (M = 61.6% and 61.9%), as shown in Fig. 7b.

Heart Disease. Editable explanations were particularly beneficial in this task (Fig. 7c). Editable participants showed strong performance (M = 89.2% and 92.8%), with large effects relative to Read-only (d = 1.99 and 1.62). CoExplain achieved similarly high faithfulness (M = 84.4% and 93.9%), also with large differences from Read-only (d = 1.56 and 1.62). In contrast, Read-only participants performed much worse overall (M = 60.5% and 74.4%) and benefited far less from explanations that were not aligned through user editing.

Across all datasets, Editable and CoExplain supported higher initial levels of user-AI faithfulness and larger gains once explanations were available, demonstrating improved understanding. These results support our hypothesis that editable explanations leverage the *generation effect* from active learning: by constructing and refining rules, users internalized the AI's behavior more effectively, amplifying the benefits of explanation access.

6.3.3 *CoExplain Improve AI Accuracy.*

As shown in Fig. 8a, the model revealed significant main effects of *XAI Type* and *Dataset* (p < .0001), with no significant interaction or effect of Edit Iterations. Read-only achieved the highest pre-trained accuracy (M = 78.9%), Editable the lowest (M = 59.8%), and CoExplain (M = 69.9%) increased the accuracy from Editable with AI enhancement (d = 1.25) and minimized the gap with data optimized Read-only (d = 0.81), improving editing accuracy without sacrificing alignment.

6.3.4 *Editable Explanations Improve Prediction Alignment.*

We found strong effects of *Dataset* and its interaction with *XAI Type* (p < .0001). Although the main effect of *XAI Type* was not significant, least-square means showed clear improvements for *Editable* (d = 0.80, M = 71.4%) and *CoExplain* (d = 1.03, M = 75.7%) over *Read-only* (M = 49.5%). Editing effort also mattered, with Iterations showing a robust effect (p < .0001).

6.3.5 *Editable Explanations improve Topological Alignment.*

Fig. 8c shows the rule-level distance measure. The model revealed a strong main effect of *XAI Type* (p < .0001) and an interaction with *Dataset* (p = .0016). Without user editing, Read-only showed the farthest topological difference from the guideline rules (M =13.74 edits), comparing with it, Editable (d = 1.92, M = 3.13) and CoExplain (d = 1.39, M = 6.64) achieved a much shorter distance, showing a closer topological alignment with the guideline rules

Together, guideline accuracy reflects alignment at the prediction level, while distance reflects alignment at the explanation level that users directly shape. Editable maximized alignment but sacrificed performance, Read-only maximized performance but ignored user knowledge, and CoExplain balanced both—achieving near-optimal accuracy with explanations that stayed closer to user rules.

6.3.6 *User Engagement.*

We analyze editing interaction logs to compare user engagement across XAI types in terms of a) the number of edit operations, b) the number of edit iterations, and c) editing time (Fig. 9).

# Edit Operations. Fig. 9a showed a significant main effect of *XAI Type* (p < .0001), with no effects of *Dataset* or the interaction. Editable required the most operations (M = 14.05), while CoExplain reduced this load (d = 1.94, M = 9.00), indicating more efficient editing with AI support.

# Edit Iterations. As shown in Fig. 9b, *XAI Type* had a strong effect (p < .0001), with no influence from *Dataset* or the interaction. Editable users needed more revision rounds (M = 3.62) than CoExplain users (d = 1.91, M = 1.59).

Editing Time. Fig. 9c similarly showed a main effect of *XAI Type* (p < .0001). Editable users spent more time (M = 7.18 mins) than CoExplain users (M = 3.36 mins), a 53% reduction with d = 2.33, demonstrating substantial savings in editing effort.

Together, these results show that while Editable required the most edit operations, iterations, and time, CoExplain significantly reduced user effort while still allowing meaningful interaction. Combined with its performance and alignment results, CoExplain provided faithful and efficient AI enhanced editing.

6.3.7 *AI-Explanation Faithfulness and Edit Outcome Analysis.*

Faithfulness quantifies how closely the distilled decision tree matches the neural network's predictions. Since a distilled tree explanation may oversimplify the neural network model, the explanation may

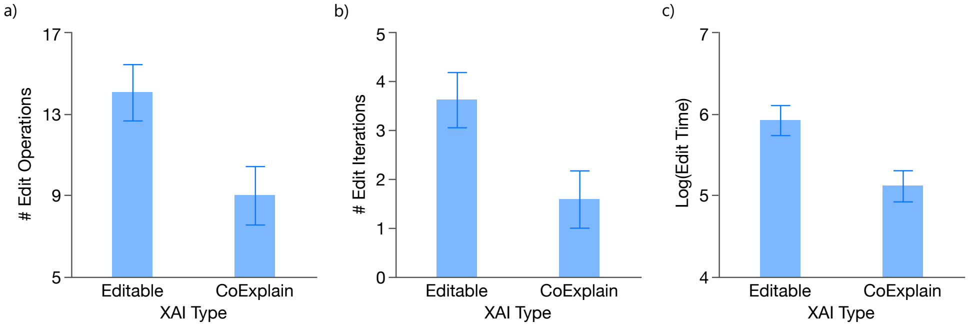


Figure 9: Results measuring user engagement in terms of: a) the number of edit operations, b) the number of edit iterations, and c) log of the time spent on editing.

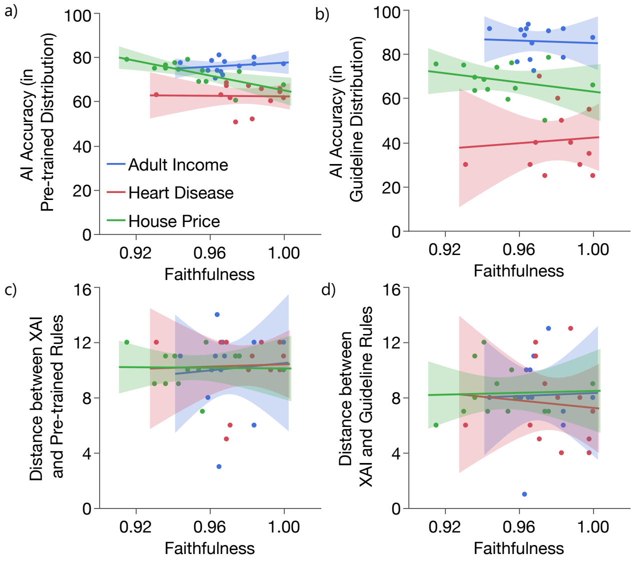


Figure 10: Analysis of edit outcomes in relation to the faithfulness of the tree explanations to the underlying model. a) AI accuracy on the pre-trained distribution, b) AI accuracy on the guideline distribution, c) TED between XAI and pre-trained rules, and d) TED between XAI and guideline rules.

be unfaithful to the model. This unfaithfulness could mislead users to editing irrelevant attributes. Hence, we investigate if faithfulness was low and whether it led to ineffective user edits. Across 350 trials from the user study, the average faithfulness was high, 97.5% for *Adult Income*, 96.8% for *House Price*, and 96.53% for *Heart Disease*. Figure 10 plots faithfulness against four editing outcomes: a) AI accuracy on the pre-trained distribution, b) AI accuracy on the guideline distribution, c) distance between the XAI rules and pre-trained rules, and d) distance between the XAI rules and guideline rules. Almost all trends are not significantly different from 0, indicating no clear effect on edit outcomes due to faithfulness. Interestingly, for House Price, accuracy decreases as Faithfulness increases, suggesting that users may be less engaged in editing a larger tree. In general, the results suggest that variation in faithfulness within the observed range did not systematically influence how well users edited the rules. Future work could study this effect by aggressively pruning the tree explanation or training a model on a significantly more complex domain.

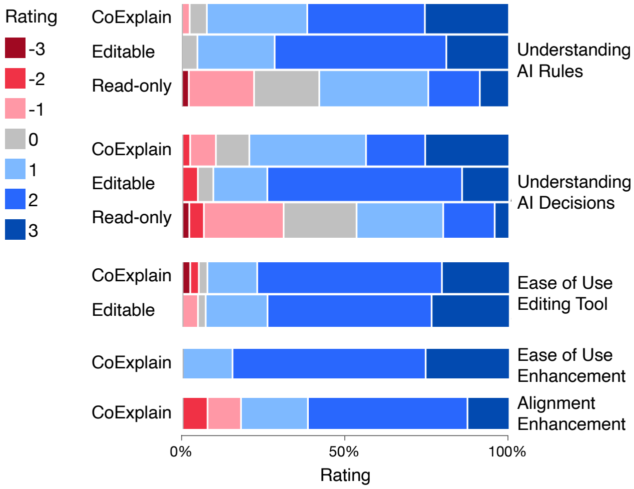


Figure 11: Perceived ratings of understanding of AI rules and decisions, ease of use, and alignment of enhancements across XAI types. AI enhancement is only available in CoExplain, Read-only does not support editing.

6.3.8 Perceived Ratings. We collected perceived ratings from participants about their understanding of the resulting AI model in terms of its rule explanations and decisions, the ease of use of the system, and the ease of use and alignment of CoExplain enhancements (Fig. 11).

For understanding the AI model, editable conditions were clearly preferred. A mixed-effects analysis revealed a significant effect of *XAI Type* on perceived understanding of AI behavior ($p = 0.0006$) and rules ($p = 0.0008$). Post-hoc contrast tests showed that both Editable ($M = 1.69$ (≈ 2 Agree) for Understanding of AI Decisions; $M = 1.86$ (≈ 2 Agree) for Understanding of AI Rules) and CoExplain ($M = 1.36$ (≈ 1 Somewhat Agree); $M = 1.77$ (≈ 2 Agree)) were rated significantly higher than Read-only ($M = 0.31$ (≈ 0 Neither Agree nor Disagree); $M = 0.64$ (≈ 1 Somewhat Agree)).

Participants generally rated the editing interface as easy to use. Most users found CoExplain's enhancement suggestions controllable and well aligned with their rules with an acceptance rate of 89.74% of enhancement edits. However, a few participants expressed dissatisfaction with certain AI suggestions, noting that they occasionally felt misaligned with their goals. C19, who was familiar with the real estate, complained about the AI on the house price prediction task as *"I think it was way too simple."*

In summary, participants perceived interactive explanations as more supportive of understanding the AI model, easier to use, and better aligned with them. These perceptions reinforce the value of engaging users in a collaborative relationship with the system, rather than limiting them to static, read-only explanations.

6.4 Summary of Results

Our results show that Editable explanations improved user–AI faithfulness, understanding, and perceived ease of use compared to Read-only explanations. The summative study demonstrated that editable explanations led to higher alignment and better understanding of the AI logic, with CoExplain further reducing editing effort through co-editing enhancements. Qualitative findings revealed that participants used editing to align the AI with their knowledge and engaged with enhancements to refine thresholds and topology under their control. Together, these results suggest that editable explanations, and CoExplain in particular, foster more effective and efficient human–AI alignment than static, read-only explanations.

7 DISCUSSION

We discuss Editable XAI and its implementation as CoExplain in terms of the scope, relation to other approaches, and the collaborative challenges of human–AI co-editing. Together these themes highlight how editable explanations create new opportunities for bi-directional alignment between humans and AI.

7.1 Scope and Generalization of CoExplain

While our implementation of **Editable XAI** as **CoExplain** targeted structured data with decision tree as the knowledge representation, we discuss how the framework of Editable XAI can be generalized.

7.1.1 Beyond Symbolic Rules to General Human Knowledge. CoExplain was designed specifically for human knowledge expressed as symbolic rules. However, the broader principle of Editable XAI is not limited to this representation. In many domains, experts encode knowledge through heuristic equations or domain-specific formulae, for example, a risk score calculated as $\text{Risk} = 0.3 \times \text{Age} + 0.5 \times \text{Blood Pressure}$, which, like rules, offer a shared medium interpretable by both humans and AI. Extending editable explanations to such mathematical expressions would broaden applicability, allowing collaboration around quantitative heuristics as well as logical conditions. While our work does not address this direction directly, research on knowledge distillation [47], neurosymbolic integration [53], and equation learner networks [69] highlight promising avenues that could inspire future methods for parsing and refining equations within editable XAI frameworks.

7.1.2 Beyond Neural Networks to Diverse AI Models. While CoExplain is implemented for neural networks, the principle of Editable XAI as using an editable representation as a shared medium for human–AI alignment is not limited to this model class. Neural networks provide high accuracy, flexible gradient-descent training, and the ability to incorporate user-specified constraints, making them well-suited for editable explanations. Editable XAI is also valuable for glass-box models, such as decision trees or decision sets [77], where users can already inspect and directly modify model components. In high-stakes domains, combining glass-box models with editable explanations can further enhance alignment, transparency, and trust [107]. For other black-box models, such as recurrent or ensemble models, Editable XAI would require identifying an appropriate intermediate representation, similar to how CoExplain uses decision tree rules, that preserves model fidelity while supporting human edits and comprehension. This suggests that the concept of Editable XAI could potentially be applied more broadly across diverse model families, beyond neural networks.

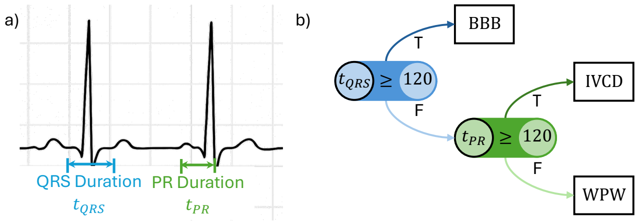


Figure 12: Use of CoExplain on ECG signal data with feature extraction. Concept features extracted using NeuroKit2 [93], decision tree constructed from MG Khan [67]. Leaf nodes denotes diagnosis, Bundle Branch Block (BBB), IntraVentricular Conduction Delay (IVCD) and Wolff-Parkinson-White syndrome (WPW).

7.1.3 Beyond Structured Data to Concept-based Explanations. Our evaluation targeted binary classification tasks on structured data, though the parsing algorithm also supports multiclass classification. Extending to regression tasks would require new rule-to-network mappings that accommodate continuous outputs. Extending to unstructured data can be done by first interpreting semantic concepts, then reasoning over them in a structured (logical) way.

With the importance of explaining in interpretable concepts [86, 129], XAI techniques for concept-based explanations can be used to infer human interpretable concepts [45, 70, 103], that are then treated as tabular data to be explained and edited with CoExplain. Recent works have integrated concept extraction and logical neural networks [27, 62], but they are not editable. Concept-based explanations assume that the concepts align with human prior knowledge, but if they are spurious, users may want to edit them [58]. Hence, future work can extend CoExplain for unstructured data by using concept-based explanations with editable concepts.

As a preliminary proof-of-concept, we demonstrate CoExplain for medical diagnosis on electrocardiogram (ECG) with ECG concept features from a time series ECG trace (see Fig. 12, modeling details in Appendix A.2). Here, we extracted concepts (e.g., QRS Duration t_{QRS}, PR Duration t_{PR}; see Fig. 12a) using the NeuroKit2 [94], and used CoExplain to train a neural network, and explain with a decision tree (Fig. 12b) based on those concept features.

7.1.4 Beyond Monolithic Model to Modular Sub-Models. We have explored the use of CoExplain to parse and distill interpretable decision trees with feedforward neural networks. To manage the cognitive for our participants, we had pruned the tree size to be no larger than a depth of 4 levels and 16 leaves. This also limited the corresponding neural networks to about 5 layers with 150 neurons in total. Yet, the technical approach is not limited to small trees or networks, and can scale to larger models. For example, an unpruned tree of depth 10 can correspond to a NN of 20, 460 neurons. Recent works knowledge distillation techniques (e.g., layer-wise

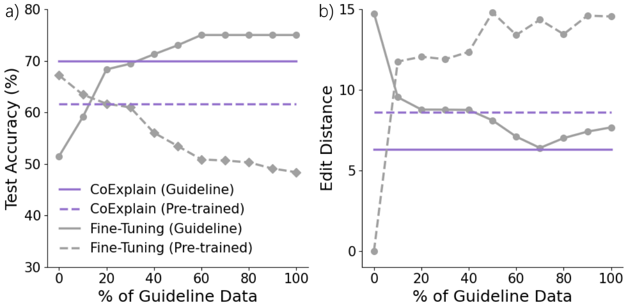


Figure 13: Comparison of CoExplain and Direct Fine-Tuning showing the trade-off between effortful human labeling and alignment to distributions.

distillation [117], soft decision trees [39], tree embeddings [80]) can be leveraged to further scale to the overall technical approach to accommodate modern deep models like DenseNet-169 [122].

In contrast, human cognition is more bounded, and this presents scalability challenges too, as observed for large expert systems [61, 83]. Hence, how could users edit a truly large and complex model? We hypothesize that abstraction [14] with modularity [86, 95, 129] could help users to interpret the model hierarchically with sub-models. Instead of editing a large monolithic model, the user can separately edit each sub-model and refine them along a shared module-module interface. This is beyond the scope of our work and should be designed and examined in future work.

7.2 Relation to Other XAI and AI Approaches

We situate Editable XAI within broader work on human–AI interaction, highlighting both its differences from and complementarities with interactive explanations, interactive machine learning and large language models.

7.2.1 Editable XAI vs. Interactive Explanations. Existing interactive explanation methods focus on enhancing user understanding by allowing queries through clarifications such as counterfactuals [96], examples [21], or contrastive cases [19, 119]. While effective for inspection, these approaches typically leave the underlying reasoning fixed, limiting users' ability to align the model with their domain knowledge. Editable XAI introduces a new level of interaction by making explanations themselves editable: users can directly reshape the decision logic or reasoning artifacts, which in turn updates the model's behavior. CoExplain exemplifies this approach, demonstrating that editable explanations can improve bi-directional alignment, allowing users not only to understand but also to correct and align model reasoning. The concept of Editable XAI is general and could be applied to other explanation techniques, suggesting a broader paradigm where explanations serve as manipulable interfaces for collaborative human–AI reasoning, especially in domains where users already possess domain expertise (e.g., medicine or finance) and seek not just to interpret but also to correct or refine the model's reasoning for alignment [14, 86].

7.2.2 Editable XAI vs. Interactive Machine Learning. Interactive machine learning (IML) approaches, such as programming by demonstration, active learning, or reinforcement learning from human feedback, improve model behavior through iterative data provision and retraining [36]. These methods are effective for expanding coverage and adapting to new tasks but typically require substantial user effort in generating labels or demonstrations [3].

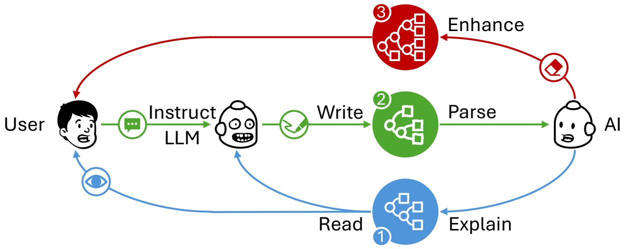


Figure 14: Extension of conceptual overview of Fig. 1 to include LLM usage in the workflow of CoExplain. This embeds the user prompting an LLM read distilled rule explanations and suggest revisions to be parsed by Editable XAI.

Editable XAI operates at a different level: rather than supplying new data, users directly modify reasoning artifacts such as rules, which can be more efficient in settings where labeling is costly, slow, or sensitive (e.g., medical diagnosis), while also providing transparency about how edits affect model logic [49].

Despite the labor intensity, fine-tuning with human labeling could achieve better performance than editing via XAI. To investigate the trade-off between effort and accuracy, we compare CoExplain with direct fine-tuning using varying amounts of guideline data on the *Heart Disease* task with 460 instances. For fine-tuning, to simulate human labeling, we used the guideline decision tree of the user study (Section 6.1.4) as the label oracle. For CoExplain, we used the average performance acquired from the user study (Section 6). Fig. 13 shows that the Fine-tuned model has less alignment toward the Guideline goal (low Test Accuracy, high Edit Distance) than CoExplain, but rises as more Guideline data is labeled. Only when 32.17% of the data (148 instances) are labeled, then would its performance match CoExplain. Assuming labeling at a rate of 6.6 seconds per instance [6], this would require 16.28 minutes of human effort, which is longer than CoExplain's 3.36 minutes.

We view Editable XAI as complementary to IML and other human-in-the-loop methods: user edits can refine reasoning structures, while data-driven techniques expand coverage in areas where rules are insufficient, and interpretable glass-box models can provide foundations for further editable adjustments [41]. Such hybrid workflows promise to balance expert alignment, efficiency, and robustness by combining the strengths of explanation editing and data-driven feedback.

7.2.3 Editable XAI and Large Language Models. Large language models (LLMs) provide alternative pathways for aligning AI with human knowledge, including fine-tuning [57], in-context learning [33], and prompt engineering [108]. While these approaches can produce flexible and context-sensitive outputs, they are inherently stochastic and may not consistently follow user-intended reasoning. LLM-generated explanations are also prone to hallucination [65], which can compromise fidelity and trustworthiness.

Editable XAI offers a complementary approach: by representing reasoning as structured, manipulable concepts, it supports predictable bi-directional alignment and allows users to directly correct or refine AI behavior. LLMs can be incorporated into this workflow by providing them with contextual information about the dataset and existing rules, for example through prompt-based descriptions or in-context examples of the structured explanations (see Appendix A.3). This enables the LLM to suggest edits, alternative rules, or explanations while staying aligned with the user's goals, explanatory schema, and data. Fig. 14 illustrates how combining LLMs with editable XAI can reduce human effort and support co-created editing goals. Fig. A.2 shows an exploratory results of the LLM-infused approach with enhanced tree explanations from CoExplain.

Future work could explore hybrid workflows to combine the flexibility of LLMs with the structured guarantees of Editable XAI, maintaining user control and leveraging LLM's suggestions.

7.3 Collaboration for Bi-Directional Alignment

Editable XAI positions explanation editing as a collaborative process. Beyond transparency, its value lies in how humans and AI negotiate shared control, adapt to each other, and co-construct reasoning over time.

7.3.1 Bringing AI into the Human Editing Loop. Editable XAI should be framed as a dialogic process, where explanations are shaped through iterative contributions from both human and AI [97]. Prior work in IML has relied heavily on human labor, such as labeling, curating, or correcting examples, while under-utilizing the AI's capacity to optimize and propose alternatives. By enabling joint edits on a shared structure, CoExplain leverages both forms of expertise: users bring domain knowledge, while AI suggests simplifications, counterexamples, or refinements grounded in data. This transforms explanations from static artifacts into evolving negotiation spaces, supporting not just model transparency but also mutual adaptation between human reasoning and machine optimization [84, 92].

7.3.2 Human Mistakes and Quality of Input. A central design challenge lies in managing the quality of human contributions. Users may provide incomplete, biased, or erroneous rules, which could propagate harmful reasoning into the system if incorporated uncritically [54, 104]. Editable XAI offers a partial safeguard by making the editable medium shared: in CoExplain, both human and AI operate on the same decision tree explanation, allowing the AI to correct mistakes through data-driven optimization, or suggest alternative edits to improve low-quality inputs. Still, additional challenges remain. One risk is overconfidence, where users trust their expertise even when it is limited, potentially leading to rigid or incorrect edits [25]. Another is over-reliance, where users defer excessively to the AI and accept its suggestions uncritically [18]. While Editable XAI can alleviate these effects by exposing reasoning structures and making corrections visible, it cannot resolve them entirely. Addressing them will also require drawing on findings from empirical XAI research on human–AI trust [100, 120] and confidence [81] to design safeguards that balance user agency with error prevention.

7.3.3 Balancing Initiative in Collaboration. Editable XAI frames collaboration through a shared editable medium, but bi-directional alignment also depends on calibrating initiative across different stages of interaction [42]. Initiative shapes not only how edits unfold, such as when humans should take the lead in editing and when the AI should propose refinements, but also how collaboration begins, such as whether rules are first authored by humans, generated by the AI, or co-constructed from partial contributions [52]. Our study focused on the human-first setting, where users defined rules and the AI adapted through optimization. Other starting points raise new opportunities and concerns. AI-initialized rules may accelerate entry into complex domains, but risk biasing users toward machine suggestions [101], while human-initialized rules foreground expertise but may overlook patterns that are only visible in data [46]. Developing mechanisms to balance initiative at both the outset and during ongoing edits remains central to positioning Editable XAI as a genuine collaboration rather than a one-sided process.

7.4 Limitations and Future Work

Our work has several limitations. First, CoExplain currently uses decision tree rules as the interaction medium, which effectively links human reasoning with model optimization, but suffers from loss of details like other forms of easily understandable explanations [121]. Future work may extend Editable XAI to additional knowledge representations, such as rule sets or heuristic non-linear equations to address broader cases.

Second, we evaluated CoExplain only on structured inputs with small models, whose attributes align well with users' deliberative reasoning. The method does not directly extend to perception tasks (e.g., vision, audio) or language reasoning, which involve innate mental processes due to stimuli or low-level, practiced skills. Future work can integrate other modalities through feature extraction and concept-based explanations (Section 7.1.3) and to larger models through modularization and abstraction (Section 7.1.4).

Third, although our user study included a heart disease task and several participants with medical backgrounds, we focused on usage by lay users. Future work should investigate how domain experts would use Editable XAI or what schemas they would wish to edit with. Different fields encode expertise in distinct forms, and enabling editing through domain-familiar representations may reduce cognitive load and support more trustworthy outcomes [86].

8 CONCLUSION

We introduced **Editable XAI** through CoExplain, a neurosymbolic framework that allows users to read, write, and enhance rule-based explanations. In a study with 43 participants, Editable XAI improved understanding and alignment compared to read-only explanations, while CoExplain further reduced effort and balanced user knowledge with near-optimal performance. These results highlight the value of writable explanations as a means of fostering bi-directional alignment and active engagement. Future work can extend Editable XAI to other explanation forms and domains, advancing more collaborative and controllable human–AI interaction.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative (Grant No. DTC-RGC-09), and by the Ministry of Education, Singapore (Grant No. MOE-T2EP20121-0010). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

REFERENCES

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems* 31 (2018).

[2] Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. 2020. Evaluating saliency map explanations for convolutional neural networks: a user study. In *Proceedings of the 25th international conference on intelligent user interfaces*. 275–285.

[3] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI magazine* 35, 4 (2014), 105–120.

[4] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104* (2017).

[5] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2019. Gradient-based attribution methods. In *Explainable AI: Interpreting, explaining and visualizing deep learning*. Springer, 169–191.

[6] Shilpa Arora, Eric Nyberg, and Carolyn Rose. 2009. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*. 18–26.

[7] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic tensor networks. *Artificial Intelligence* 303 (2022), 103649.

[8] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software* 149 (2019), 101–137.

[9] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, et al. 2019. Ilastik: interactive machine learning for (bio) image analysis. *Nature methods* 16, 12 (2019), 1226–1232.

[10] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. 2017. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 298–308.

[11] Jacob Bien and Robert Tibshirani. 2011. Prototype selection for interpretable classification. (2011).

[12] Alberto Blanco-Justicia and Josep Domingo-Ferrer. 2019. Machine learning explainability through comprehensible decision trees. In *International cross-domain conference for machine learning and knowledge extraction*. Springer, 15–26.

[13] Jessica Y Bo, Pan Hao, and Brian Y Lim. 2024. Incremental xai: Memorable understanding of ai with incremental explanations. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–17.

[14] Angie Boggust, Hyemin Bang, Hendrik Strobelt, and Arvind Satyanarayan. 2025. Abstraction Alignment: Comparing Model-Learned and Human-Encoded Conceptual Relationships. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[15] Olcay Boz. 2002. Extracting decision trees from trained neural networks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 456–461.

[16] Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. 2017. *Classification and regression trees*. Chapman and Hall/CRC.

[17] Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M Drucker, Ashish Kapoor, and Patrice Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 105–112.

[18] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-computer Interaction* 5, CSCW1 (2021), 1–21.

[19] Zana Buçinca, Siddharth Swaroop, Amanda E Paluch, Finale Doshi-Velez, and Krzysztof Z Gajos. 2025. Contrastive explanations that anticipate human misconceptions can improve human decision-making skills. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–25.

[20] Federico Cabitza, Caterina Fregosi, Andrea Campagner, and Chiara Natali. 2024. Explanations considered harmful: the impact of misleading explanations on accuracy in hybrid human-ai decision making. In *World conference on explainable artificial intelligence*. Springer, 255–269.

[21] Carrie J Cai, Jonas Jongejan, and Jess Holbrook. 2019. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th international conference on intelligent user interfaces*. 258–262.

[22] Mohammad Chegini, Jürgen Bernard, Philip Berger, Alexei Sourin, Keith Andrews, and Tobias Schreck. 2019. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Visual Informatics* 3, 1 (2019), 9–17.

[23] Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'connell, Terrance Gray, F Maxwell Harper, and Haiyi Zhu. 2019. Explaining decision-making algorithms through UI: Strategies to help non-expert stakeholders. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–12.

[24] Chun-Wei Chiang and Ming Yin. 2021. You'd better stop! Understanding human reliance on machine learning models under covariate shift. In *Proceedings of the 13th ACM Web Science Conference 2021*. 120–129.

[25] Leah Chong, Guanglu Zhang, Kosa Goucher-Lambert, Kenneth Kotovsky, and Jonathan Cagan. 2022. Human confidence in artificial intelligence and in themselves: The evolution and impact of confidence on adoption of AI advice. *Computers in Human Behavior* 127 (2022), 107018.

[26] Tsung-Nan Chou. 2019. An explainable hybrid model for bankruptcy prediction based on the decision tree and deep neural network. In *2019 IEEE 2nd international conference on knowledge innovation and invention (ICKII)*. IEEE, 122–125.

[27] Gabriele Ciravegna, Pietro Barbiero, Francesco Giannini, Marco Gori, Pietro Lió, Marco Maggini, and Stefano Melacci. 2023. Logic explained networks. *Artificial Intelligence* 314 (2023), 103822.

[28] Mark Craven and Jude Shavlik. 1995. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems* 8 (1995).

[29] Maxwell Crouse, Ibrahim Abdelaziz, Bassem Makni, Spencer Whitehead, Cristina Cornelio, Pavan Kapanipathi, Kavitha Srinivas, Veronika Thost, Michael Witbrock, and Achille Fokoue. 2021. A deep reinforcement learning approach to first-order logic theorem proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6279–6287.

[30] Geoff Cumming. 2013. *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge.

[31] Elizabeth M Daly, Massimiliano Mattetti, Öznur Alkan, and Rahul Nair. 2021. User driven model adjustment via boolean rule explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5896–5904.

[32] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2018. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management science* 64, 3 (2018), 1155–1170.

[33] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[34] Artur S d'Avila Garcez, Luís C Lamb, and Dov M Gabbay. 2009. *Neural-symbolic learning systems*. Springer.

[35] Upol Ehsan, Brent Harrison, Larry Chan, and Mark O Riedl. 2018. Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 81–87.

[36] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 39–45.

[37] Gerhard Fischer. 2012. End user development and meta-design: foundations for cultures of participation. In *End-user computing, development, and software engineering: new challenges*. IGI Global, 202–226.

[38] Andrew Frank. 2010. UCI machine learning repository. *http://archive. ics. uci. edu/ml* (2010).

[39] Nicholas Frosst and Geoffrey Hinton. 2017. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* (2017).

[40] Artur d'Avila Garcez, Tarek R Besold, Luc De Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C Lamb, Risto Miikkulainen, and Daniel L Silver. 2015. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*.

[41] Brandon L Garrett and Cynthia Rudin. 2023. Interpretable algorithmic forensics. *Proceedings of the National Academy of Sciences* 120, 41 (2023), e2301842120.

[42] Simret Araya Gebreegziabher, Yukun Yang, Elena L Glassman, and Toby Jia-Jun Li. 2025. Supporting Co-Adaptive Machine Teaching through Human Concept Learning and Cognitive Theories. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[43] Bhavya Ghai, Q Vera Liao, Yunfeng Zhang, Rachel Bellamy, and Klaus Mueller. 2021. Explainable active learning (xal) toward ai explanations as interfaces for machine teachers. *Proceedings of the ACM on human-computer interaction* 4, CSCW3 (2021), 1–28.

[44] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of context-dependent applications through trigger-action rules. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 2 (2017), 1–33.

[45] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in neural information processing systems* 32 (2019).

[46] Oihane Gómez-Carmona, Diego Casado-Mansilla, Diego Lopez-de Ipina, and Javier García-Zubia. 2024. Human-in-the-loop machine learning: Reconceptualizing the role of the user in interactive approaches. *Internet of Things* 25 (2024), 101048.

[47] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision* 129, 6 (2021), 1789–1819.

[48] Bin Guo, Daqing Zhang, and Michita Imai. 2011. Toward a cooperative programming framework for context-aware applications. *Personal and ubiquitous*

computing 15 (2011), 221–233.

[49] Lijie Guo, Elizabeth M Daly, Oznur Alkan, Massimiliano Mattetti, Owen Cornec, and Bart Knijnenburg. 2022. Building trust in interactive machine learning via user contributed interpretable rules. In *Proceedings of the 27th international conference on intelligent user interfaces*. 537–548.

[50] Harlfoxem. 2016. House Sales in King County, USA. https://www.kaggle.com/harlfoxem/housesalesprediction.

[51] Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831* (2020).

[52] Jessica He, Stephanie Houde, and Justin D Weisz. 2025. Which contributions deserve credit? perceptions of attribution in human-ai co-creation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[53] Melanie Hilario. 2013. An overview of strategies for neurosymbolic integration. *Connectionist-Symbolic Integration* (2013), 13–35.

[54] Martin Hilbert. 2012. Toward a synthesis of cognitive biases: how noisy information processing can bias human decision making. *Psychological bulletin* 138, 2 (2012), 211.

[55] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *International conference on machine learning*. PMLR, 597–606.

[56] Steven Houben, Connie Golsteijn, Sarah Gallacher, Rose Johnson, Saskia Bakker, Nicolai Marquardt, Licia Capra, and Yvonne Rogers. 2016. Physikit: Data engagement through physical ambient visualizations in the home. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 1608–1619.

[57] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[58] Lijie Hu, Chenyang Ren, Zhengyu Hu, Hongbin Lin, Cheng-Long Wang, Zhen Tan, Weimin Lyu, Jingfeng Zhang, Hui Xiong, and Di Wang. 2025. Editable Concept Bottleneck Models. 267 (13–19 Jul 2025), 24678–24726.

[59] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318* (2016).

[60] Kelli D Humbird, J Luc Peterson, and Ryan G McClarren. 2018. Deep neural network initialization with decision trees. *IEEE transactions on neural networks and learning systems* 30, 5 (2018), 1286–1295.

[61] Peter Jackson. 1986. Introduction to expert systems. (1986).

[62] Rishabh Jain, Gabriele Ciravegna, Pietro Barbiero, Francesco Giannini, Davide Buffelli, and Pietro Lio. 2022. Extending logic explained networks to text classification. *arXiv preprint arXiv:2211.09732* (2022).

[63] Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. 1988. Heart disease. *UCI machine learning repository* 10 (1988), C52P4X.

[64] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852* (2023).

[65] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys* 55, 12 (2023), 1–38.

[66] Mark T Keane and Barry Smyth. 2020. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In *International Conference on Case-Based Reasoning*. Springer, 163–178.

[67] M Gabriel Khan. 2008. *Rapid ECG interpretation*. Springer.

[68] Been Kim, Cynthia Rudin, and Julie Shah. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. *Advances in neural information processing systems* 27 (2014).

[69] Samuel Kim, Peter Y Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. 2020. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE transactions on neural networks and learning systems* 32, 9 (2020), 4166–4177.

[70] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. *International conference on machine learning*, 5338–5348.

[71] R Krishnan, G Sivakumar, and P Bhattacharya. 1999. Extracting decision trees from trained neural networks. *Pattern recognition* 32, 12 (1999).

[72] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*. 126–137.

[73] Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In *2013 IEEE Symposium on visual languages and human centric computing*. IEEE, 3–10.

[74] Sean Kulinski and David I Inouye. 2023. Towards explaining distribution shifts. In *International Conference on Machine Learning*. PMLR, 17931–17952.

[75] Samuli Laato, Miika Tiainen, AKM Najmul Islam, and Matti Mäntymäki. 2022. How to explain AI systems to end users: a systematic literature review and research agenda. *Internet Research* 32, 7 (2022), 1–31.

[76] Vivian Lai, Samuel Carton, Rajat Bhatnagar, Q Vera Liao, Yunfeng Zhang, and Chenhao Tan. 2022. Human-ai collaboration via conditional delegation: A case study of content moderation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–18.

[77] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1675–1684.

[78] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 131–138.

[79] Min Hun Lee, Daniel P Siewiorek, Asim Smailagic, Alexandre Bernardino, and Sergi Bermúdez Bermúdez i Badia. 2021. A human-ai collaborative approach for clinical decision making on rehabilitation assessment. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–14.

[80] Jiawei Li, Yiming Li, Xingchun Xiang, Shu-Tao Xia, Siyi Dong, and Yun Cai. 2020. Tnt: An interpretable tree-network-tree learning framework using knowledge distillation. *Entropy* 22, 11 (2020), 1203.

[81] Jingshu Li, Yitian Yang, Q Vera Liao, Junti Zhang, and Yi-Chieh Lee. 2025. As Confidence Aligns: Understanding the Effect of AI Confidence on Human Self-confidence in Human-AI Decision Making. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–16.

[82] Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. *arXiv preprint arXiv:1906.06298* (2019).

[83] Shu-Hsien Liao. 2005. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications* 28, 1 (2005), 93–103.

[84] Yi-Chi Liao, Paul Streli, Zhipeng Li, Christoph Gebhardt, and Christian Holz. 2025. Continual Human-in-the-Loop Optimization. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–26.

[85] Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. 2006. End-user development: An emerging paradigm. In *End user development*. Springer, 1–8.

[86] Brian Y Lim, Joseph P Cahaly, Chester YF Sng, and Adam Chew. 2025. Diagrammatization and Abduction to Improve AI Interpretability With Domain-Aligned Explanations for Medical Diagnosis. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–25.

[87] Han Liu, Vivian Lai, and Chenhao Tan. 2021. Understanding the effect of out-of-distribution examples and interactive explanations on human-ai decision making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–45.

[88] Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624* (2021).

[89] Wei-Yin Loh. 2011. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 1, 1 (2011), 14–23.

[90] Mitja Luštrek, Matjaž Gams, Sanda Martinčić-Ipšić, et al. 2016. What makes classification trees comprehensible? *Expert Systems with Applications* 62 (2016), 333–346.

[91] Shuai Ma, Ying Lei, Xinru Wang, Chengbo Zheng, Chuhan Shi, Ming Yin, and Xiaojuan Ma. 2023. Who should i trust: Ai or myself? leveraging human and ai correctness likelihood to promote appropriate trust in ai-assisted decision-making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[92] Prashan Madumal, Tim Miller, Frank Vetere, and Liz Sonenberg. 2018. Towards a grounded dialog model for explainable artificial intelligence. *arXiv preprint arXiv:1806.08055* (2018).

[93] Dominique Makowski, Tam Pham, Zen J Lau, Jan C Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and SH Annabel Chen. 2021. NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior research methods* 53, 4 (2021), 1689–1696.

[94] Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. 2021. NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods* 53, 4 (feb 2021), 1689–1696. https://doi.org/10.3758/s13428-020-01516-y

[95] Hitoshi Matsuyama, Nobuo Kawaguchi, and Brian Y Lim. 2023. Iris: Interpretable rubric-informed segmentation for action quality assessment. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 368–378.

[96] Jacqueline Michelle Metsch, Anna Saranti, Alessa Angerschmid, Bastian Pfeifer, Vanessa Klemt, Andreas Holzinger, and Anne-Christin Hauschild. 2024. CLARUS: an interactive explainable AI platform for manual counterfactuals in graph neural networks. *Journal of biomedical informatics* 150 (2024), 104600.

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

[98] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2023. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review* 56, 4 (2023), 3005–3054.

[99] Duy T Nguyen, Kathryn E Kasmarik, and Hussein A Abbass. 2020. Towards interpretable ANNs: An exact transformation to multi-class multivariate decision

trees. *arXiv preprint arXiv:2003.04675* (2020).

[100] Saumya Pareek, Sarah Schömbs, Eduardo Velloso, and Jorge Goncalves. 2025. " It's Not the AI's Fault Because It Relies Purely on Data": How Causal Attributions of AI Decisions Shape Trust in AI Systems. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[101] Pat Pataranutaporn, Chayapatr Archiwaranguprok, Samantha WT Chan, Elizabeth Loftus, and Pattie Maes. 2025. Synthetic human memories: Ai-edited images and videos can implant false memories and distort recollection. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[102] David Piorkowski, Inge Vejsbjerg, Owen Cornec, Elizabeth M Daly, and Öznur Alkan. 2023. AIMEE: An Exploratory Study of How Rules Support AI Developers to Explain and Edit Models. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023), 1–25.

[103] Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. 2023. Concept-based explainable artificial intelligence: A survey. *Comput. Surveys* (2023).

[104] Emily Pronin. 2007. Perception and misperception of bias in human judgment. *Trends in cognitive sciences* 11, 1 (2007), 37–43.

[105] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " 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*. 1135–1144.

[106] Ryan Riegel, Alexander Gray, Francois Luus, Naweed Khan, Ndivhuwo Makondo, Ismail Yunus Akhalwaya, Haifeng Qian, Ronald Fagin, Francisco Barahona, Udit Sharma, et al. 2020. Logical neural networks. *arXiv preprint arXiv:2006.13155* (2020).

[107] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence* 1, 5 (2019), 206–215.

[108] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927* (2024).

[109] Franco Scarselli and Ah Chung Tsoi. 1998. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural networks* 11, 1 (1998), 15–37.

[110] Gregor PJ Schmitz, Chris Aldrich, and Francois S Gouws. 1999. ANN-DT: an algorithm for extraction of decision trees from artificial neural networks. *IEEE Transactions on Neural Networks* 10, 6 (1999), 1392–1401.

[111] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[112] Hua Shen, Tiffany Knearem, Reshmi Ghosh, Kenan Alkiek, Kundan Krishna, Yachuan Liu, Ziqiao Ma, Savvas Petridis, Yi-Hao Peng, Li Qiwei, et al. 2024. Towards bidirectional human-ai alignment: A systematic review for clarifications, framework, and future directions. *arXiv preprint arXiv:2406.09264* (2024).

[113] Hua Shen, Tiffany Knearem, Reshmi Ghosh, Michael Xieyang Liu, Andrés Monroy-Hernández, Tongshuang Wu, Diyi Yang, Yun Huang, Tanushree Mitra, Yang Li, et al. 2025. Bidirectional Human-AI Alignment: Emerging Challenges and Opportunities. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–6.

[114] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345* (2020).

[115] Norman J Slamecka and Peter Graf. 1978. The generation effect: Delineation of a phenomenon. *Journal of experimental Psychology: Human learning and Memory* 4, 6 (1978), 592.

[116] Kacper Sokol and Peter Flach. 2020. One explanation does not fit all: The promise of interactive explanations for machine learning transparency. *KI-Künstliche Intelligenz* 34, 2 (2020), 235–250.

[117] Jie Song, Haofei Zhang, Xinchao Wang, Mengqi Xue, Ying Chen, Li Sun, Dacheng Tao, and Mingli Song. 2021. Tree-like decision distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13488–13497.

[118] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[119] Stefano Teso and Kristian Kersting. 2019. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 239–245.

[120] Sterre Van Arum, Hüseyin Uğur Genç, Dennis Reidsma, and Armağan Karahanoğlu. 2025. Selective Trust: Understanding Human-AI Partnerships in Personal Health Decision-Making Process. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–21.

[121] Lynn Vonder Haar, Timothy Elvira, and Omar Ochoa. 2023. An analysis of explainability methods for convolutional neural networks. *Engineering Applications of Artificial Intelligence* 117 (2023), 105606.

[122] Adarsh Vulli, Parvathaneni Naga Srinivasu, Madipally Sai Krishna Sashank, Jana Shafi, Jaeyoung Choi, and Muhammad Fazal Ijaz. 2022. Fine-tuned DenseNet-169 for breast cancer metastasis prediction using FastAI and 1-cycle policy. *Sensors* 22, 8 (2022), 2988.

[123] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. 2020. PTB-XL, a large publicly available electrocardiography dataset. *Scientific data* 7, 1 (2020), 1–15.

[124] Lin Wang and Kuk-Jin Yoon. 2021. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence* 44, 6 (2021), 3048–3068.

[125] Merlin C Wittrock. 1992. Generative learning processes of the brain. *Educational psychologist* 27, 4 (1992), 531–541.

[126] Tongshuang Wu, Daniel S Weld, and Jeffrey Heer. 2019. Local decision pitfalls in interactive machine learning: An investigation into feature selection in sentiment analysis. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 4 (2019), 1–27.

[127] Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. 2021. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open* 2 (2021), 14–35.

[128] Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing* 18, 6 (1989), 1245–1262.

[129] Wencan Zhang and Brian Y Lim. 2022. Towards relatable explainable AI with the perceptual process. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–24.

[130] Simon Zhuang and Dylan Hadfield-Menell. 2020. Consequences of misaligned AI. *Advances in Neural Information Processing Systems* 33 (2020), 15763–15773.

A APPENDIX

A.1 Data Partitioning

To provide participants with a unified baseline of knowledge across the three scenarios, we partitioned each dataset into two distributions: a **Guideline Distribution** and a **pre-trained Distribution**. The guideline distribution was used as the knowledge source for participants to construct their decision rules. This mimics the partial, textbook-like knowledge that cannot be directly applied to real-world practice. In contrast, the pre-trained distribution was used to train the AI model and was not revealed to participants, representing real-world data that cannot be fully anticipated from the guideline distribution.

This partitioning created a deliberate misalignment between participants' reasoning and the AI model's prediction behavior. Our design follows prior work on tackling Out-of-Distribution (OOD) challenges in human-AI collaboration [24, 74, 76, 87, 88]. We split by gender for *Adult Income* and *Heart Disease*, and by age (below/above 50) for *House Price*. Distribution shifts were quantified via feature-wise Wasserstein distances and label KL divergence. *Adult Income* showed moderate shifts (Wasserstein = 0.4844, KL = 0.1412). *House Price* showed the largest feature shift (Wasserstein = 0.8587) but minimal label shift (KL = 0.0188), reflecting age-related housing differences without major price change. *Heart Disease* showed the smallest feature shift (Wasserstein = 0.3003) but the largest label shift (KL = 0.3092), reflecting gender-based prevalence differences with stable feature distributions. Together, these datasets cover different types of distribution shifts.

A.2 Demonstration of Applying CoExplain on ECG Signal Data with Feature Extraction

We implement and generalize our CoExplain explanation method to ECG signal data as a demonstration with feature extraction.

A.2.1 Dataset. PTB-XL [123] contains 21837 multi-labeled 10-second ECG records gathered from 18885 patients. In total, PTB-XL has 71 labels consisting of diagnostic labels describing specific CVDs associated with the ECG, form labels describing ECG's morphology, and cardiac axis labels.

A.2.2 Feature Extraction. We extract interpretable ECG features using NeuroKit2 [93], following standard clinical and signal-processing conventions. The pipeline computes time-domain intervals (e.g., RR, PR, QRS, QT), frequency-domain heart-rate–variability (HRV) metrics, and morphology-based measurements (e.g., amplitudes and segment deviations), all of which can be directly incorporated into CoExplain as editable concepts. Details of extracted features can be found in Table 2.

A.2.3 Domain Rules. We constructed the decision tree rules for ECG diagnosis from *Rapid ECG Interpretation* by MG Khan [67]. ECG diagnosis are organized into 9 interconnected steps, we converted the 9 steps flowcharts into 9 decision tree rules, each used to train a specific module of the ECG diagnosis network. Fig. A.1 shows an example of the ECG diagnosis rules.

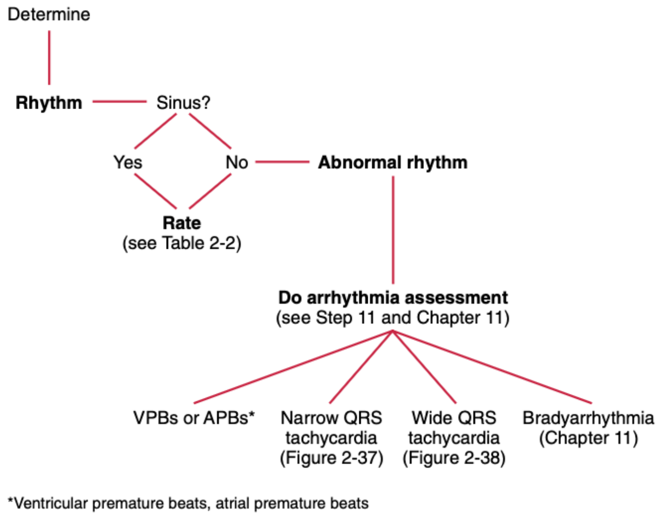

Determine
Rhythm
Sinus?
Yes
No
Abnormal rhythm
Rate
(see Table 2-2)
Do arrhythmia assessment
(see Step 11 and Chapter 11)
VPBs or APBs*
Narrow QRS tachycardia (Figure 2-37)
Wide QRS tachycardia (Figure 2-38)
Bradyarrhythmia (Chapter 11)
*Ventricular premature beats, atrial premature beats

Figure A.1: Example of ECG step rules used in *Rapid ECG Interpretation* (MG Khan, 2008, p. 36) [67]. Step-by-step method for accurate ECG interpretation. Step 1: Assess rhythm and rate.

A.2.4 Model Organization. For each of the 9 steps, we have one neural network regularized by one decision tree rule to predict certain diagnosis with specific concept input features, steps dependent of others steps are interconnected and takes the previous steps' output as input.

A.2.5 Training and Predicting. We initialize the step modules with the parsing algorithm, and treat the ECG rules from [93] as the source of human edit, and regularize the training of the neural networks with them. The prediction task performed was a multi-label task and a total of 21 diagnoses were considered, models trained with Adam optimizer for 50 epochs. After testing on the test set, CoExplain have an accuracy of 92.02% with an AUROC of 0.8360.

Table 2: ECG Features, Abbreviations, and Explanations

Feature Name	Abbreviation	Explanation
Heart Rate	HR	Heart Rate of the patient
Bradycardia	BRAD	Whether the patient has bradycardia (HR < 60 bpm)
Tachycardia	TACH	Whether the patient has tachycardia (HR > 100 bpm)
Sinus	SINUS	Whether the rhythm is sinus: Each P wave in lead II should be positive AND precedes a QRS complex
RR interval range	RR_DIFF	max R–R interval − min R–R interval
PR duration	PR_DUR	Duration of the PR segment
Prolonged PR	LPR	Whether the PR interval is prolonged
QRS duration	QRS_DUR	Duration of the QRS complex
Prolonged QRS	LQRS	Whether the QRS complex is prolonged
Prolonged QRS for WPW	LQRS_WPW	Whether the QRS complex is prolonged by WPW's standards
Short PR	SPR	Whether the PR interval is shortened
ST segment amplitude	ST_AMP_x	Mean amplitude of ST segment in lead x
ST Elevation	STE_x	Whether the ST segment is elevated in lead x
ST Depression	STD_x	Whether the ST segment is depressed in lead x
Poor R-wave Progression	PRWP	Whether R waves are not within desired ranges for at least one lead in V1–V4
Q wave duration	Q_DUR_x	Duration of the Q wave in lead x
Q wave amplitude	Q_AMP_x	Amplitude of Q wave in lead x
Pathological Q wave	PATH_Q_x	Whether the Q wave in lead x is pathological
P wave duration	P_DUR_x	Duration of P wave in lead x
P wave amplitude	P_AMP_x	Amplitude of P wave in lead x
Prolonged P wave	LP_x	Whether the P wave is prolonged in lead x
Peaked P wave	PEAK_P_x	Whether the P wave is peaked (high amplitude) in lead x
Age	AGE	Age of the patient
Old age	AGE_OLD	Whether the patient's age is greater than 30
Male	MALE	Whether the patient is male
R wave amplitude	R_AMP_x	Amplitude of R wave in lead x
S wave amplitude	S_AMP_x	Amplitude of S wave in lead x
R/S Ratio	RS_RATIO_x	Ratio between amplitudes of R and S waves in lead x
Peaked R wave	PEAK_R_x	Whether the R wave is peaked in lead x
Deep S wave	DEEP_S_x	Whether the S wave is deep (low amplitude) in lead x
Dominant R wave	DOM_R_x	Whether R wave amplitude is greater than that of the S wave
Dominant S wave	DOM_S_x	Whether S wave amplitude is greater than that of the R wave
T wave amplitude	T_AMP_x	Amplitude of T wave in lead x
Inverted T wave	INVT_x	Whether the T wave is inverted in lead x
Sum of QRS	QRS_SUM	The QRS area above the baseline minus the QRS area below
Positive QRS	POS_QRS_x	The QRS is positive in lead x
Normal cardiac axis	NORM_AXIS	Whether the patient has a normal cardiac axis
Left axis deviation	LAD	Whether the patient's cardiac axis deviates toward the left
Right axis deviation	RAD	Whether the patient's cardiac axis deviates toward the right

A.3 Prompt and Example from Comparing LLM with CoExplain

Prompt. Do not use any previous chat's memory; only provide an output in the input form of JSONLogic. Suggest an enhancement on this rule to improve accuracy on the attached dataset. While staying faithful and making changes minimal, you are not required to preserve the rule exactly. You may refine thresholds, modify structure, prune branches, or extend the rule with attributes from the dataset.

Input.

```
{"if":[
  {">":["marital-status",0.5]},
  {"if":[
    {">":["education-level",13]},
    "high",
    {"if":[{">":["working-hours",36]},"high","low"]}
  ]},
  {"if":[
    {">":["investment-gain",0.5]},
    {"if":[
      {">":["education-level",13]},
      {"if":[{">":["age",30]},"high","low"]},
      "low"
    ]},
    {"if":[{">":["working-hours",36]},"high","low"]}
  ]}
]}
```

Output.

```
{"if":[
  {">":["marital-status",0.5]},
  {"if":[
    {">":["education-level",12]},
    "high",
    {"if":[
      {">":["working-hours",35]},
      {"if":[{">":["age",28]},"high","low"]},
      "low"
    ]}
  ]},
  {"if":[
    {">":["investment-gain",0]},
    {"if":[
      {">":["education-level",12]},
      {"if":[
        {">":["working-hours",40]},
        "high",
        {"if":[{">":["age",32]},"high","low"]}
      ]},
      "low"
    ]},
    {"if":[{">":["hours-per-week",38]},"high","low"]}
  ]}
]}
```

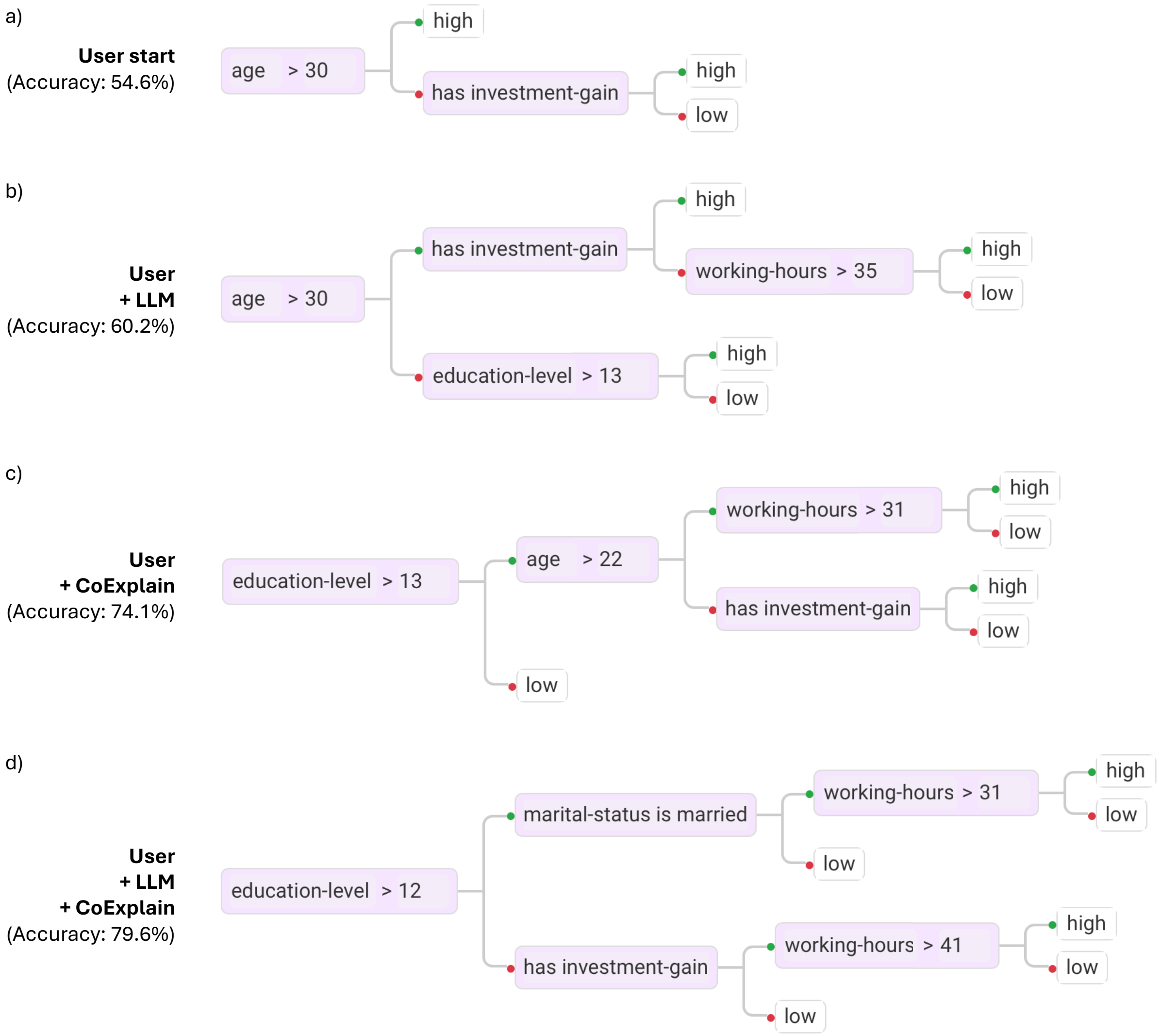


Figure A.2: Comparison of decision trees based on the Adult Income dataset as a demonstration. user a) writes initial rules, b) rewrites with LLM-prompted rule edits, c) receives rules after (a) enhanced by CoExplain, and d) receives rules after (b) enhanced by CoExplain.

A.4 Edit History Snapshots

Fig. A.3 shows the difference on human editing effort between Editable and CoExplain, we report the averaged structure editing operations, threshold editing operations and total editing operations per edit iteration. Results suggest that most editing operations are on structure, while threshold edits comes in later iterations as minor refinements, Editable users adjust their threshold more frequently than CoExplain users. Editable users takes more iterations than CoExplain users to finalize their model, and continue to make minor changes on thresholds during later iterations.

Fig. A.4 shows an editing history snapshot from the user study with C15, who spent 4 iterations with CoExplain to reach a final decision tree rule. From the edit history, user started with a self-created decision tree rule of relatively low accuracy of 66.02%, and used the CoExplain enhancement to refine the rules, AI edits are shown on the middle bar of the screenshot. For the first iteration, AI made 4 edits on C15's creation, and increased the accuracy to 71.77%, without drastic change on user's original creation. C15 accepted CoExplain's editing and choose to further refine on the AI's creation, together with AI enhancement, the accuracy gradually increased to 73.77%, and C15 finalized the model after the third iteration.

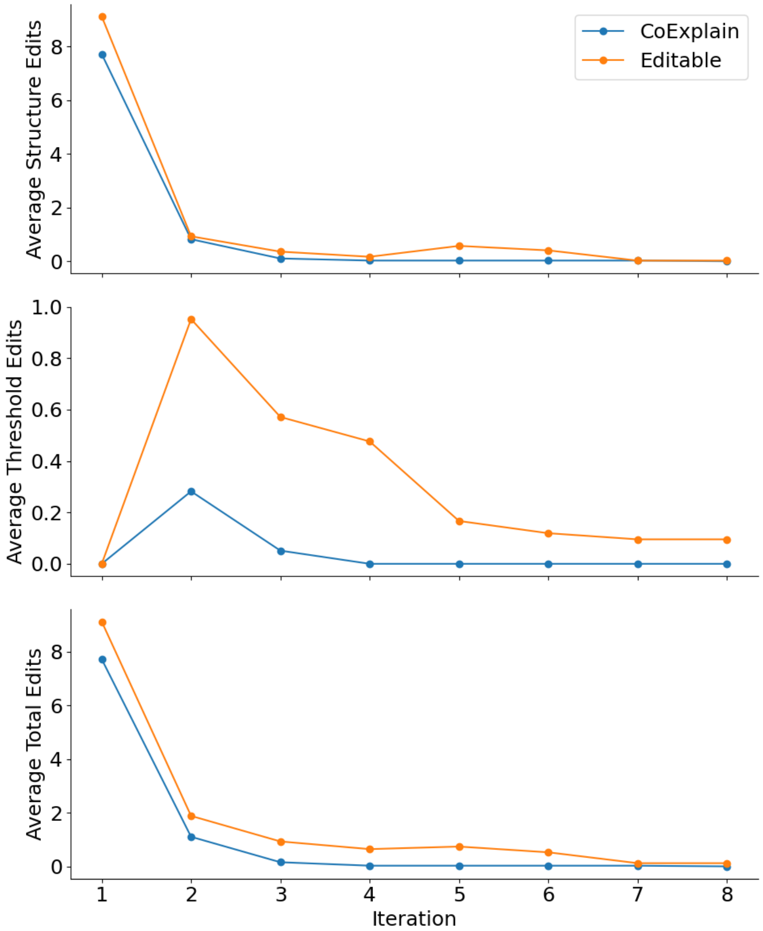


Figure A.3: Human editing effort from Editable and CoExplain

Iteration 1:

Left – User initial creation with **66.02%** accuracy.

Right – AI enhanced version with **71.77%** accuracy.

AI edits shown on middle bar "Edit Operations".

Iteration 2:

Left – User accepted previous AI suggestion and done further edits, with **68.64%** accuracy.

Right – AI enhanced version based on user edits with **71.89%** accuracy.

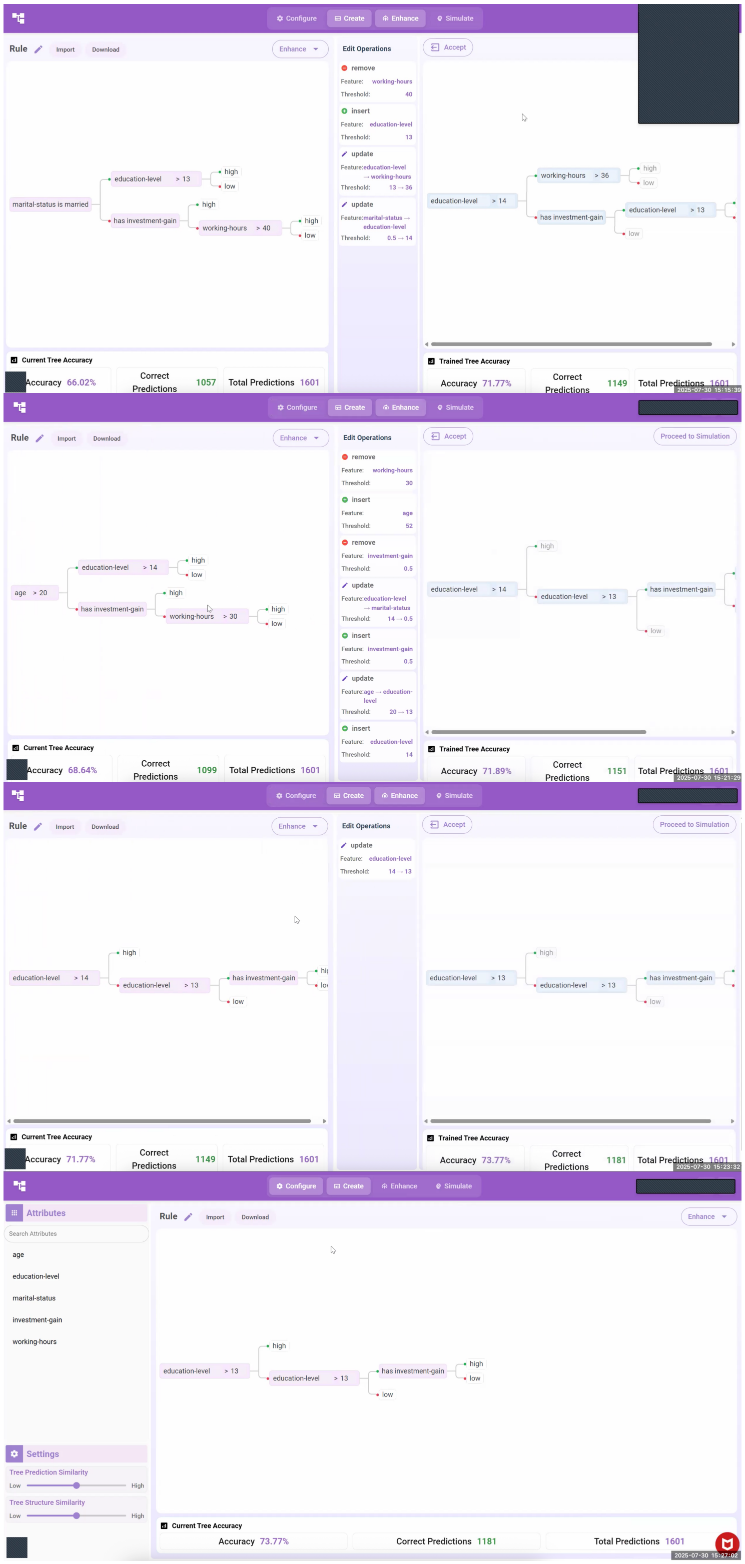
Iteration 3:

Left – User accepted previous AI suggestion and done further edits, with **71.77%** accuracy.

Right – AI enhanced version based on user edits with **73.77%** accuracy.

Iteration 4:

Left – User accepted previous AI suggestion and set it as final version.



[redacted]

Edit Operations

remove
Feature: working-hours
Threshold: 40

insert
Feature: education-level
Threshold: 13

update
Feature: education-level → working-hours
Threshold: 13 → 36

update
Feature: marital-status → education-level
Threshold: 0.5 → 14

Current Tree Accuracy
[redacted] Accuracy 66.02% | Correct Predictions 1057 | Total Predictions 1601

Trained Tree Accuracy
Accuracy 71.77% | Correct Predictions 1149 | Total Predictions 1601

2025-07-30 15:15:39

[redacted]

Edit Operations

remove
Feature: working-hours
Threshold: 30

insert
Feature: age
Threshold: 52

remove
Feature: investment-gain
Threshold: 0.5

update
Feature: education-level → marital-status
Threshold: 14 → 0.5

insert
Feature: investment-gain
Threshold: 0.5

update
Feature: age → education-level
Threshold: 20 → 13

insert
Feature: education-level
Threshold: 14

Current Tree Accuracy
[redacted] Accuracy 68.64% | Correct Predictions 1099 | Total Predictions 1601

Trained Tree Accuracy
Accuracy 71.89% | Correct Predictions 1151 | Total Predictions 1601

2025-07-30 15:21:29

[redacted]

Edit Operations

update
Feature: education-level
Threshold: 14 → 13

Current Tree Accuracy
[redacted] Accuracy 71.77% | Correct Predictions 1149 | Total Predictions 1601

Trained Tree Accuracy
Accuracy 73.77% | Correct Predictions 1181 | Total Predictions 1601

2025-07-30 15:23:32

[redacted]

Current Tree Accuracy
Accuracy 73.77% | Correct Predictions 1181 | Total Predictions 1601

[redacted]

2025-07-30 15:27:02

Figure A.4: Edit history snapshot from C15 on adult income task.

A.5 Read-only Explanations

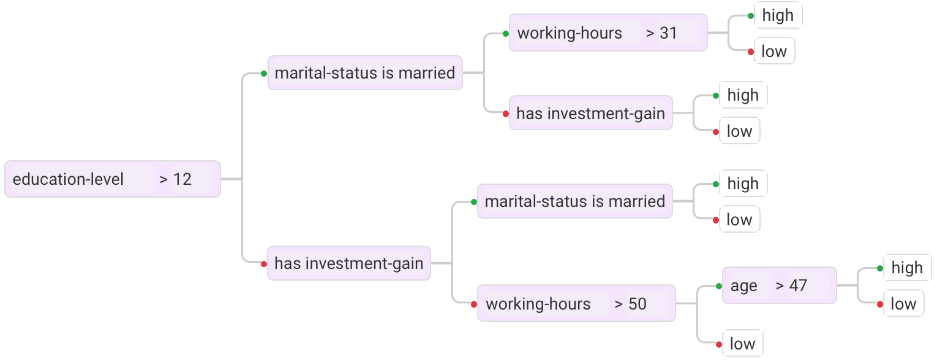


Figure A.5: Read-only explanation for Adult Income Prediction Task.

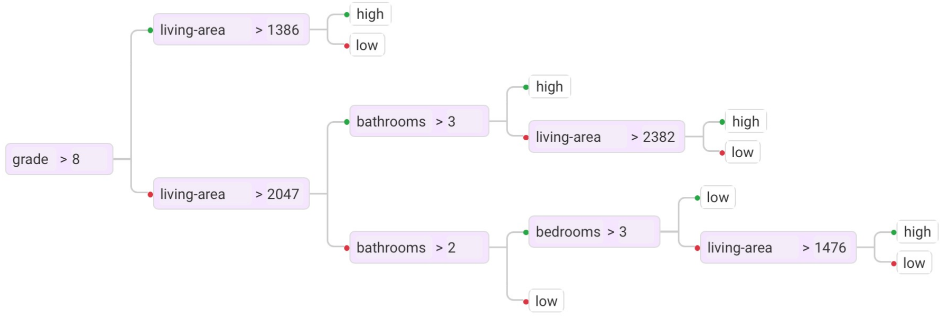


Figure A.6: Read-only explanation for House Price Prediction Task.

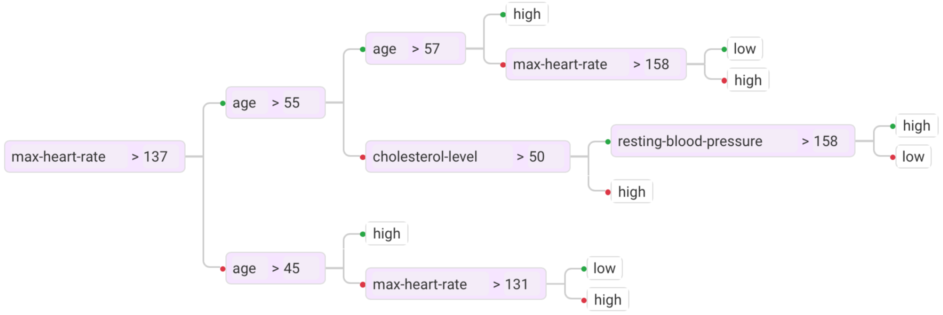


Figure A.7: Read-only explanation for Heart Disease Prediction Task.

A.6 Examples of User Created Decision Tree Rules

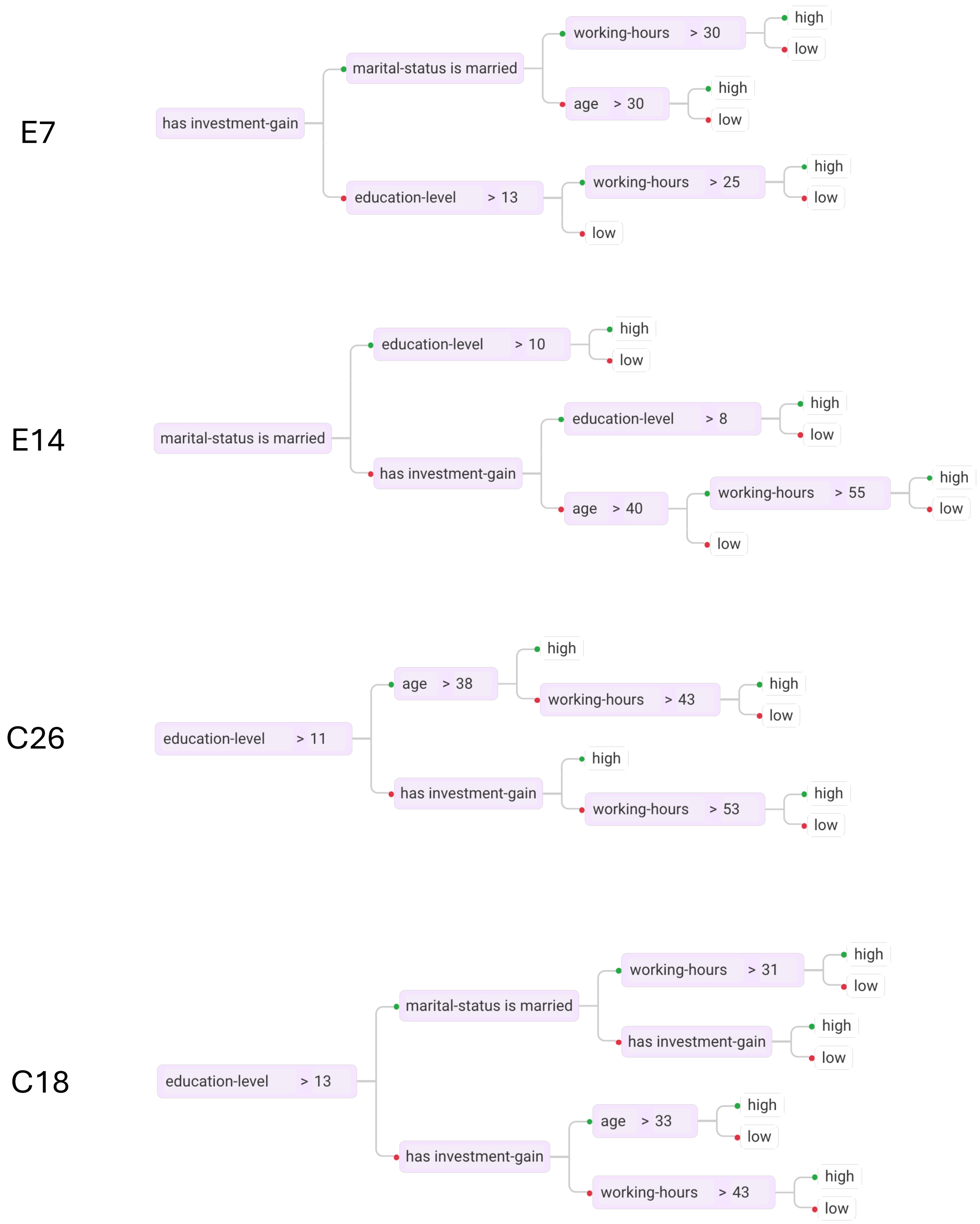


Figure A.8: Examples of user created decision tree rules on the Adult Income Prediction Task

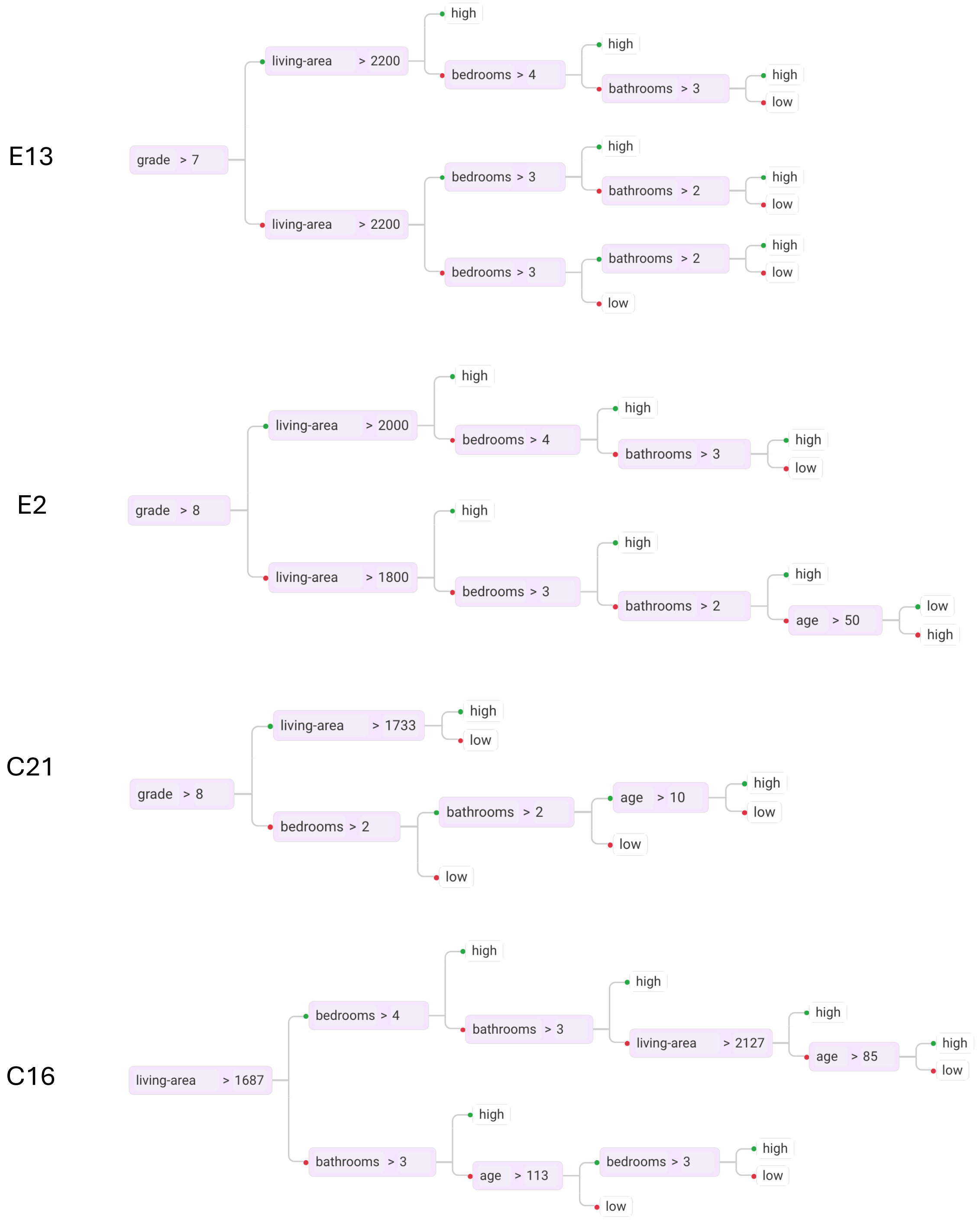


Figure A.9: Examples of user created decision tree rules on the House Price Prediction Task

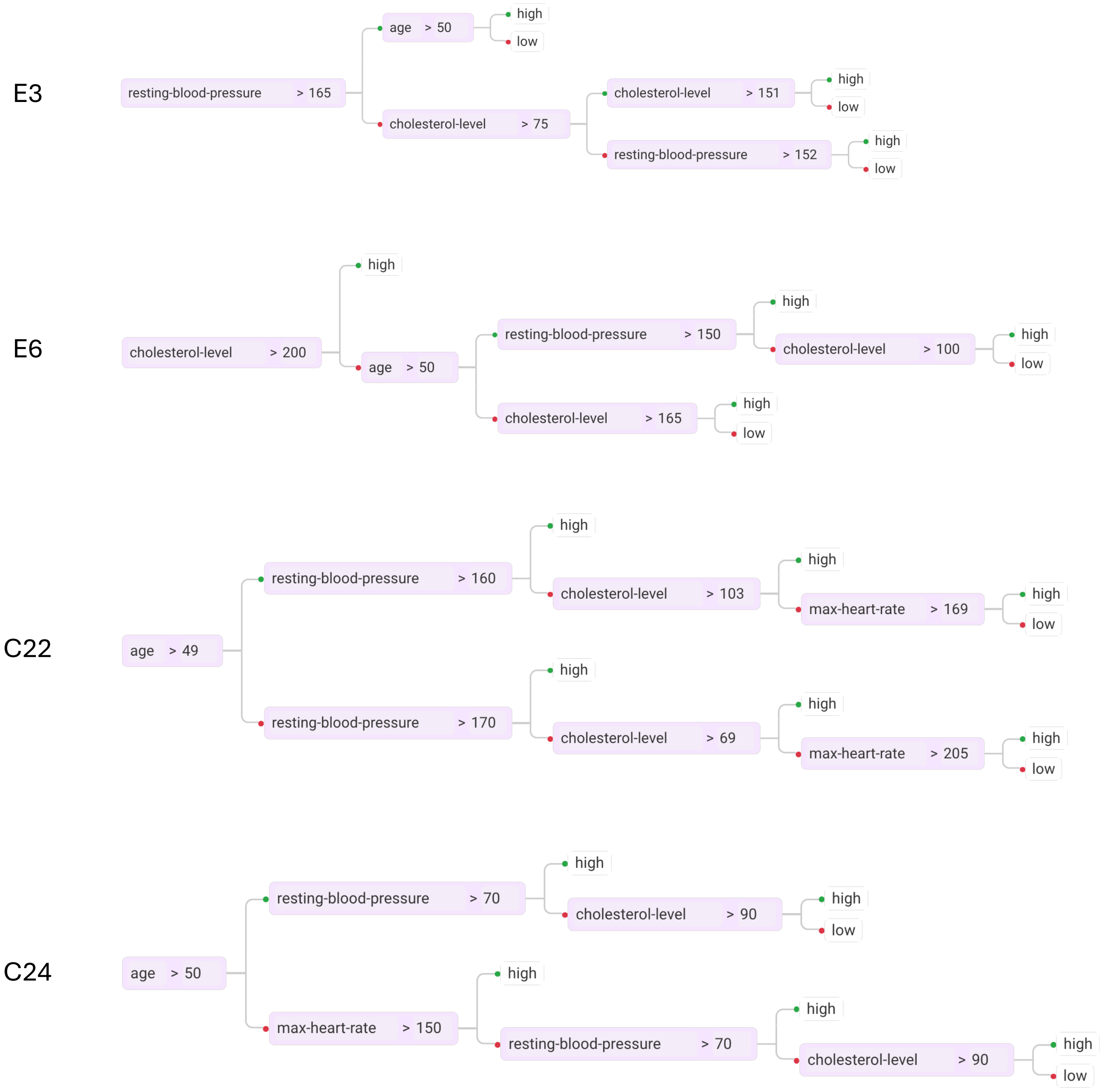
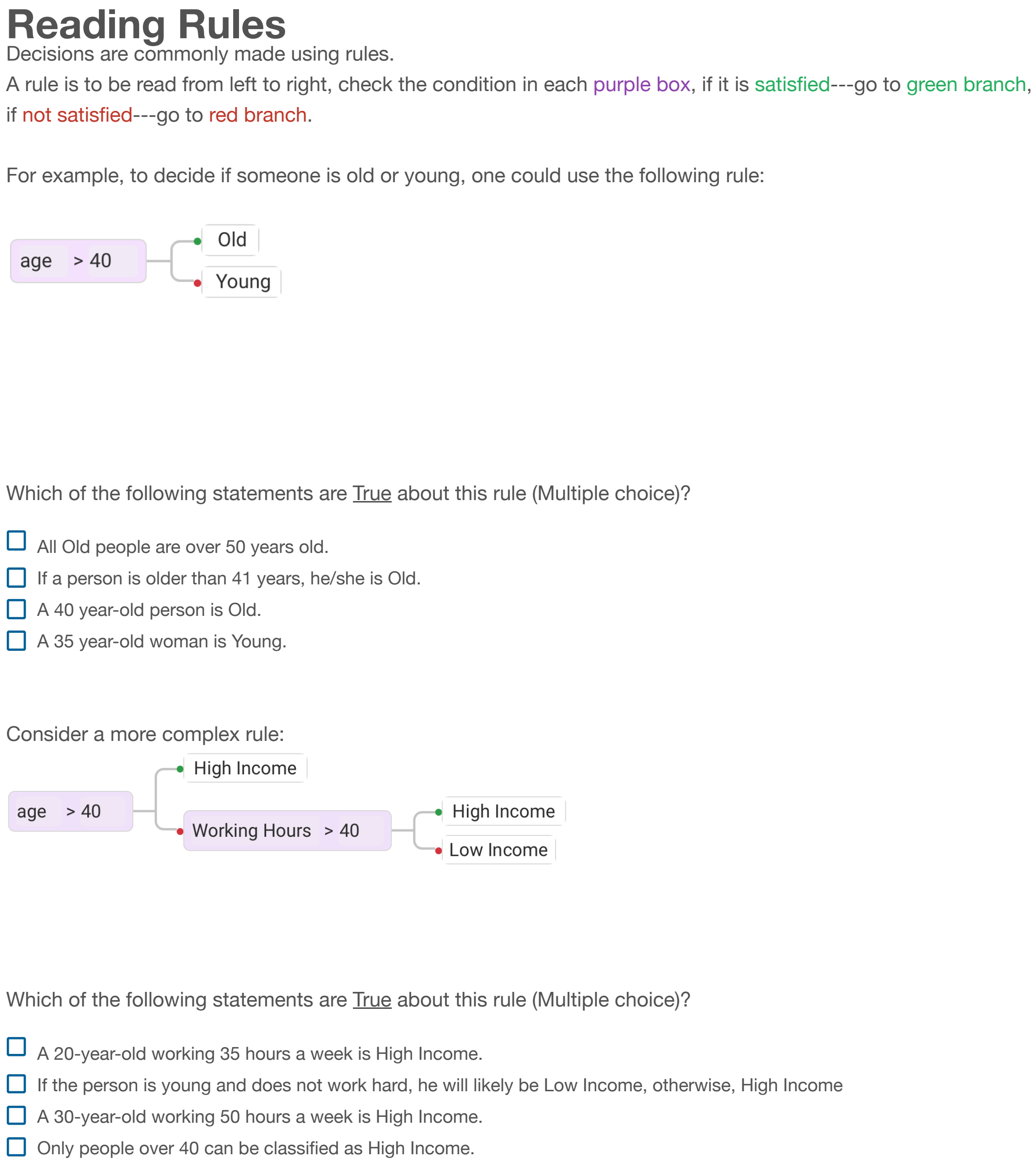


Figure A.10: Examples of user created decision tree rules on the Heart Disease Prediction Task

A.7 Survey for the User Study



Reading Rules

Decisions are commonly made using rules.

A rule is to be read from left to right, check the condition in each purple box, if it is satisfied---go to green branch, if not satisfied---go to red branch.

For example, to decide if someone is old or young, one could use the following rule:

Which of the following statements are True about this rule (Multiple choice)?

- □ All Old people are over 50 years old.
- □ If a person is older than 41 years, he/she is Old.
- □ A 40 year-old person is Old.
- □ A 35 year-old woman is Young.

Consider a more complex rule:

Which of the following statements are True about this rule (Multiple choice)?

- □ A 20-year-old working 35 hours a week is High Income.
- □ If the person is young and does not work hard, he will likely be Low Income, otherwise, High Income
- □ A 30-year-old working 50 hours a week is High Income.
- □ Only people over 40 can be classified as High Income.

Figure A.11: Tutorial on reading the rules and screening questions to check the users' interpretation.

AI Interface Tutorial

Your goal is to observe and understand how an **AI System** makes predictions, then use your understanding to anticipate its behavior.
This tutorial is to familiarize you with the AI interface you will be using to achieve this goal.

This Interface consists 4 stage, as shown on the top bar: **Configure**, **Create**, **Enhance**, **Simulate** ①**.**

Configure is where you can locate the attributes and configure the AI assistant:

1. Attribute List ②: You can click on the attribute in this list to pick up and use in your rules, click again to drop.
2. Enhance Settings ③, you can set the AI assistant's enhancement in 2 aspects:
    1. Prediction Similarity: How similar should AI enhanced rules be with yours in terms of prediction behavior.
    2. Structure Similarity: How similar should AI enhanced rules be with yours in terms of rule flowchart.

Create is where you can create and edit the rules with the following components:

1. Rule Canvas ④:
    1. Inside the purple box, you can drop an attribute on the first box, then type your value for this attribute in the second box.
    2. The two white box represents the decisions, type High/Low to make a decision, or drop an attribute here to add a rule.
2. AI Enhancement ⑤, you are assisted by an AI to enhance your rules, it offers 2 types of enhancements:
    1. Enhance Rule Flowchart: AI will enhance your rule structure, revise your attributes and their position.
    2. Enhance Value: AI will keep your rule structure, revise your value for the attributes.

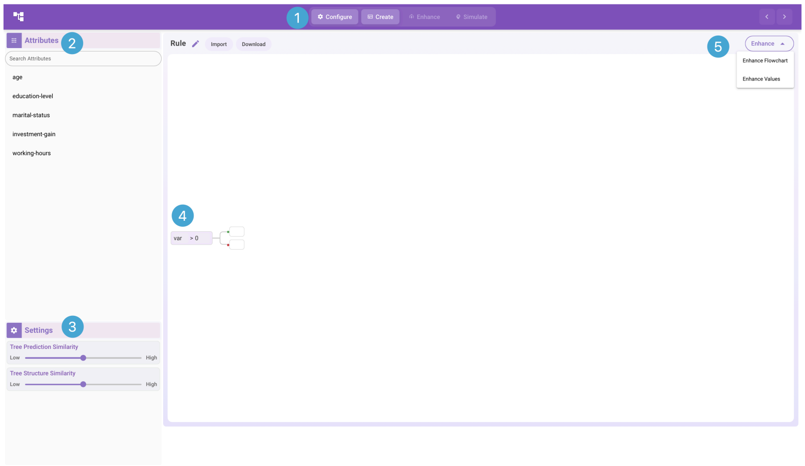
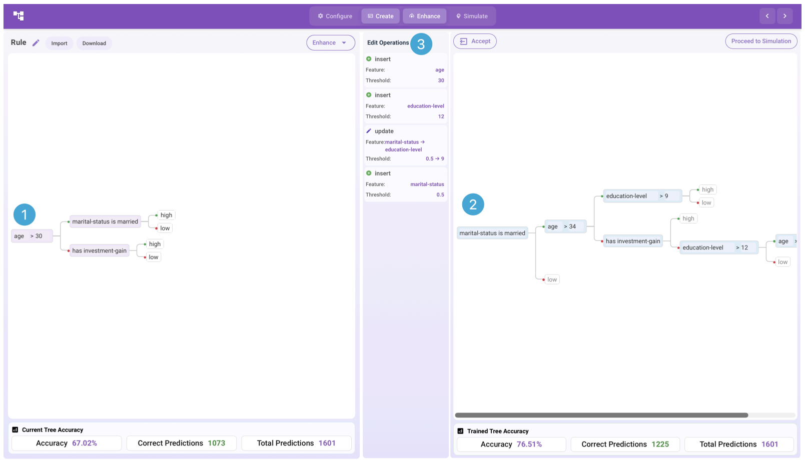


Figure A.12: Tutorial on the AI interface.

Enhance is where you can check the AI enhancement, and compare it with your own:

1. Your Rules are shown on the left ①.
2. **AI Rules** are shown on the right ②.
3. AI edits are shown in the middle ③**.**



Edit Operations

- insert
  Feature: age
  Threshold: 30
- insert
  Feature: education-level
  Threshold: 12
- update
  Feature: marital-status → education-level
  Threshold: 0.5 → 9
- insert
  Feature: marital-status
  Threshold: 0.5

Current Tree Accuracy

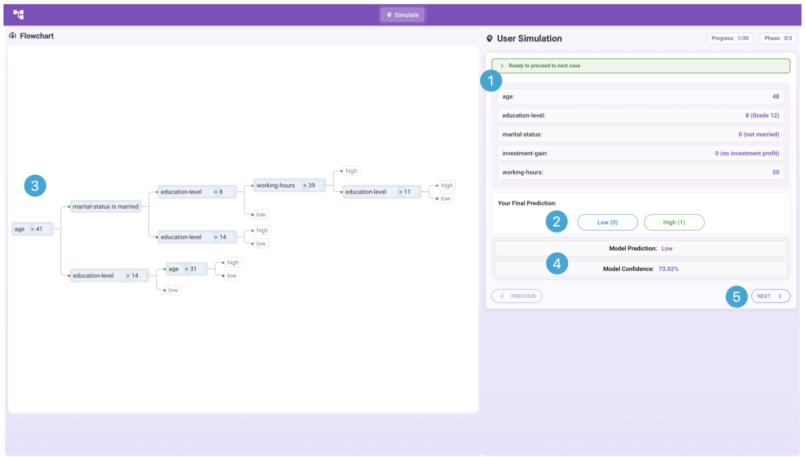
Accuracy 67.02% | Correct Predictions 1073 | Total Predictions 1601

Trained Tree Accuracy

Accuracy 76.51% | Correct Predictions 1225 | Total Predictions 1601

Simulate is where you are required to observe and anticipate the AI behavior on different cases:

1. View a case with **Attributes** ① and estimate the AI Prediction by selecting Low/High ②. This is **not** your prediction.
2. The **AI Rules** ③ will be available on the left, view the **AI Rules**, and estimate the AI prediction, again.
3. View the **AI prediction** ④, and review your answer to learn for future cases ⑤.
4. Go back to **Create** stage to revise the **AI Rules** as needed, and estimate the AI prediction, again.



User Simulation

Progress: 1/30 | Phase: 3/3

Ready to proceed to next case

age: 48

education-level: 8 (Grade 12)

marital-status: 0 (not married)

investment-gain: 0 (no investment profit)

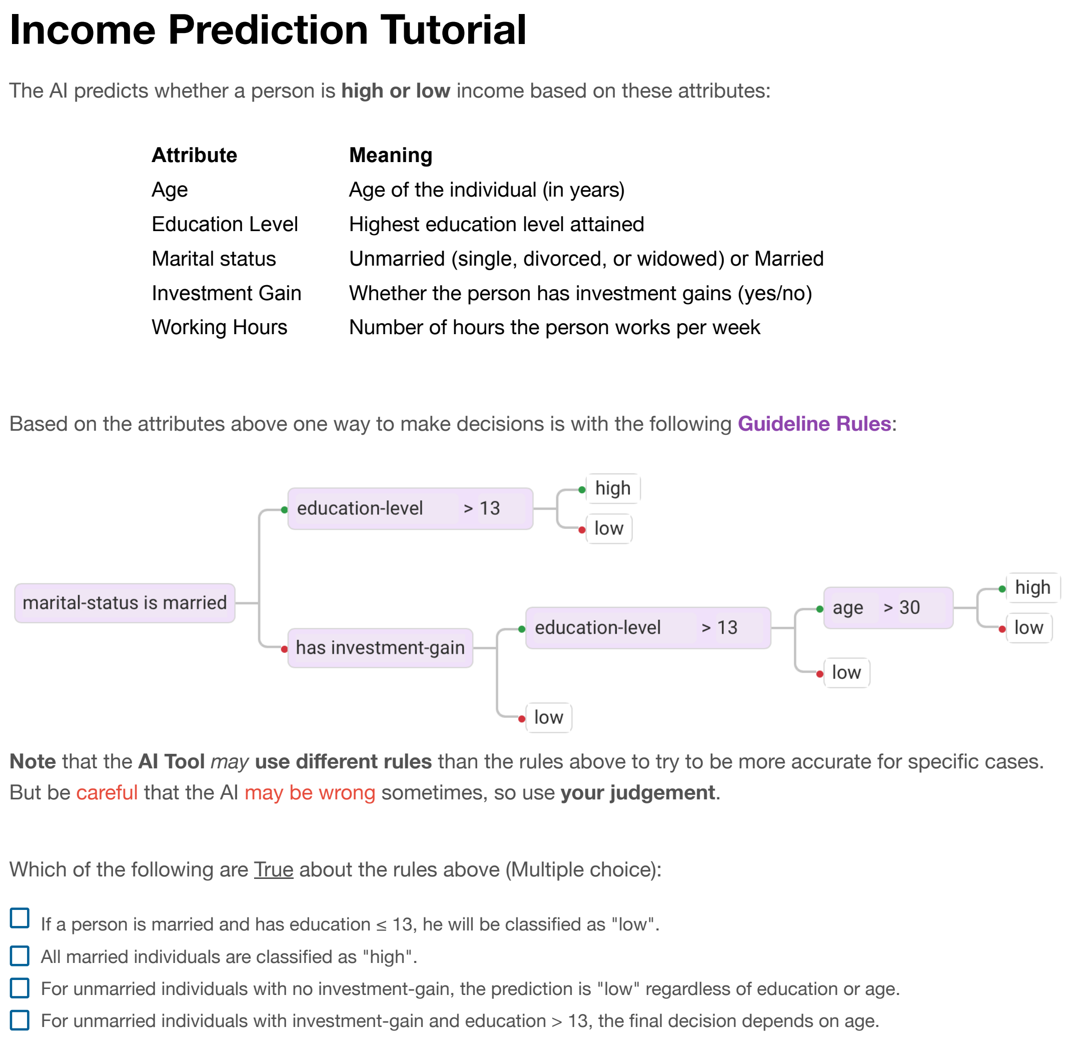
working-hours: 50

Your Final Prediction:

Model Prediction: Low

Model Confidence: 73.02%

Figure A.12: (continued)



Income Prediction Tutorial

The AI predicts whether a person is **high or low** income based on these attributes:

Attribute	**Meaning**
Age	Age of the individual (in years)
Education Level	Highest education level attained
Marital status	Unmarried (single, divorced, or widowed) or Married
Investment Gain	Whether the person has investment gains (yes/no)
Working Hours	Number of hours the person works per week

Based on the attributes above one way to make decisions is with the following **Guideline Rules**:

Note that the **AI Tool** *may* **use different rules** than the rules above to try to be more accurate for specific cases. But be careful that the AI may be wrong sometimes, so use **your judgement**.

Which of the following are True about the rules above (Multiple choice):

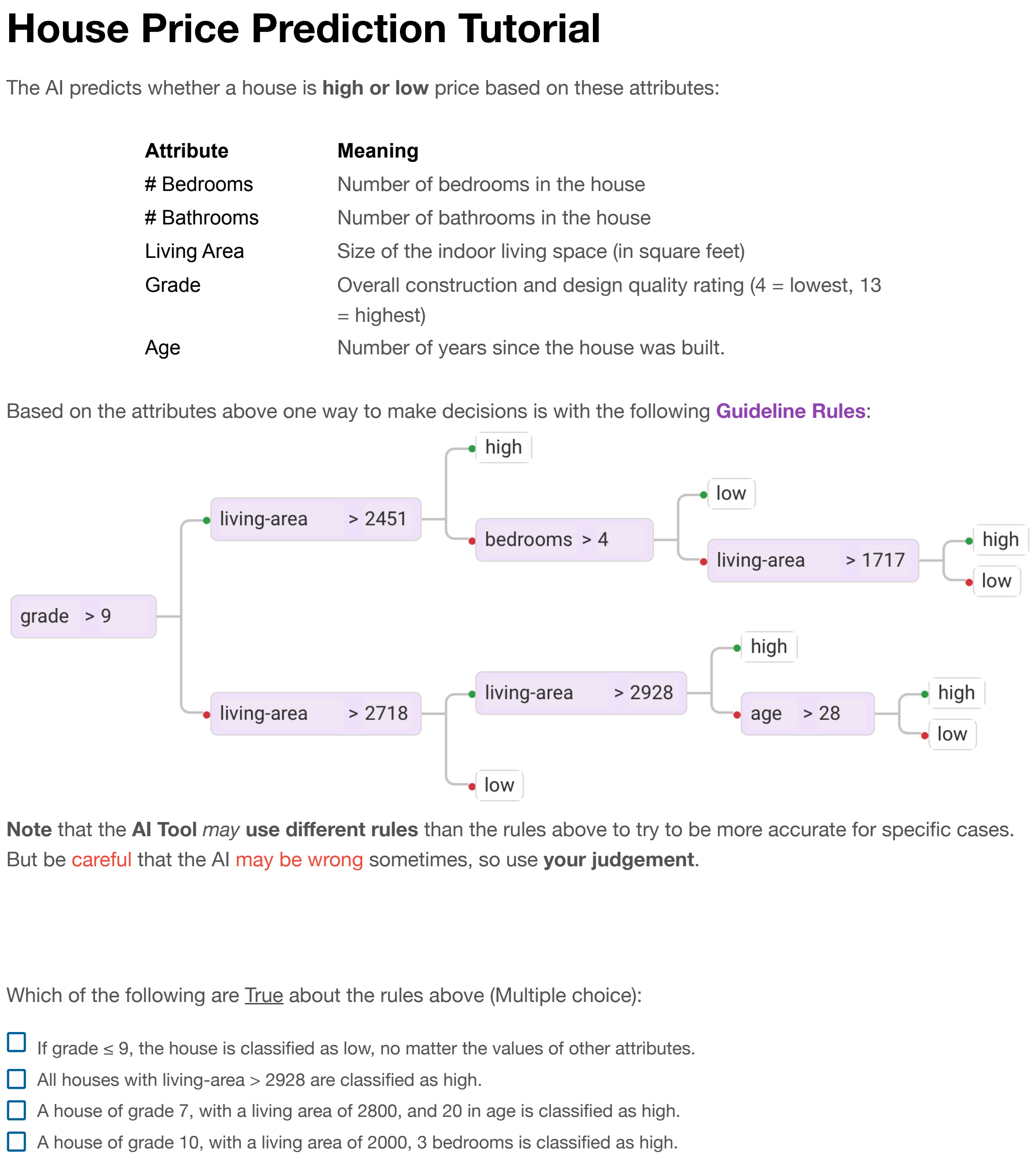
□ If a person is married and has education ≤ 13, he will be classified as "low".

□ All married individuals are classified as "high".

□ For unmarried individuals with no investment-gain, the prediction is "low" regardless of education or age.

□ For unmarried individuals with investment-gain and education > 13, the final decision depends on age.

Figure A.13: Tutorial on the Adult Income Prediction Task and screening questions to check the users' interpretation.



House Price Prediction Tutorial

The AI predicts whether a house is **high or low** price based on these attributes:

Attribute	**Meaning**
# Bedrooms	Number of bedrooms in the house
# Bathrooms	Number of bathrooms in the house
Living Area	Size of the indoor living space (in square feet)
Grade	Overall construction and design quality rating (4 = lowest, 13 = highest)
Age	Number of years since the house was built.

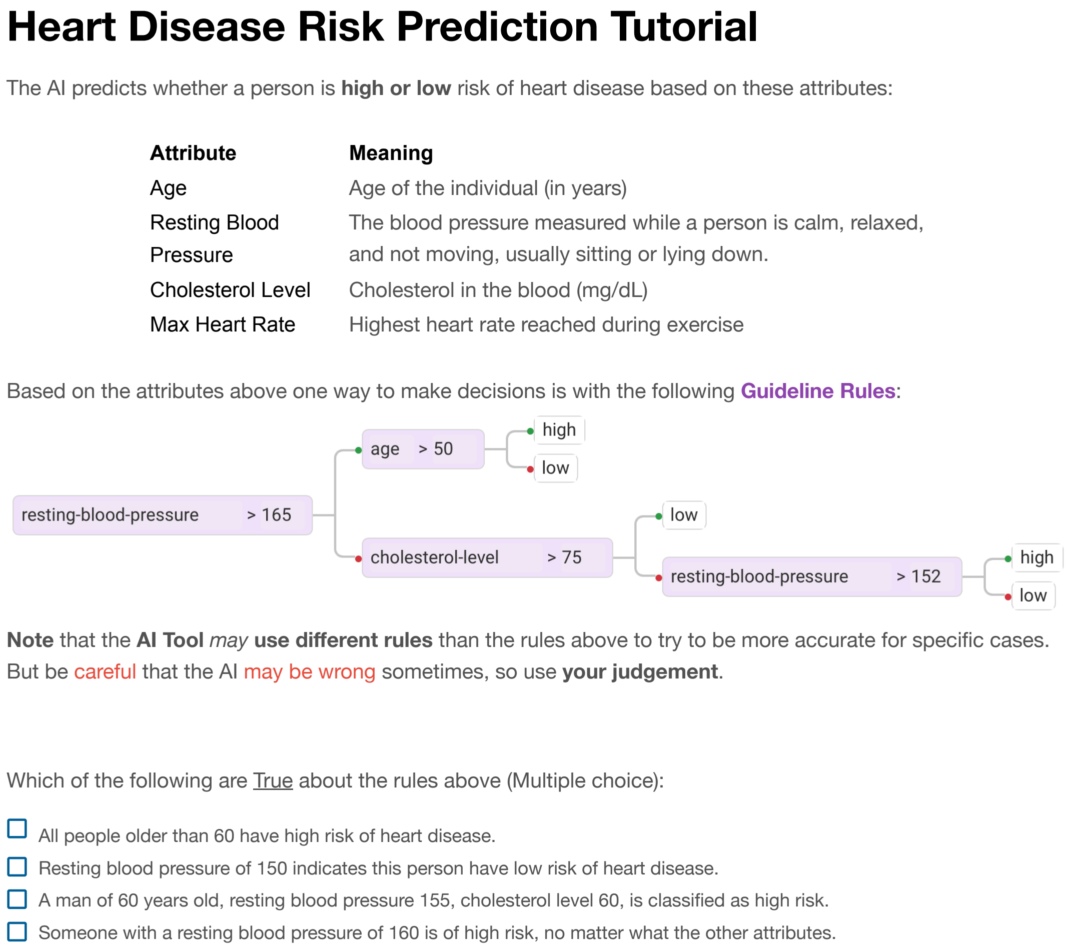
Based on the attributes above one way to make decisions is with the following **Guideline Rules**:

Note that the **AI Tool** *may* **use different rules** than the rules above to try to be more accurate for specific cases.
But be careful that the AI may be wrong sometimes, so use **your judgement**.

Which of the following are True about the rules above (Multiple choice):

□ If grade ≤ 9, the house is classified as low, no matter the values of other attributes.
□ All houses with living-area > 2928 are classified as high.
□ A house of grade 7, with a living area of 2800, and 20 in age is classified as high.
□ A house of grade 10, with a living area of 2000, 3 bedrooms is classified as high.

Figure A.14: Tutorial on the House Price Prediction Task and screening questions to check the users' interpretation.



Heart Disease Risk Prediction Tutorial

The AI predicts whether a person is **high or low** risk of heart disease based on these attributes:

Attribute	**Meaning**
Age	Age of the individual (in years)
Resting Blood Pressure	The blood pressure measured while a person is calm, relaxed, and not moving, usually sitting or lying down.
Cholesterol Level	Cholesterol in the blood (mg/dL)
Max Heart Rate	Highest heart rate reached during exercise

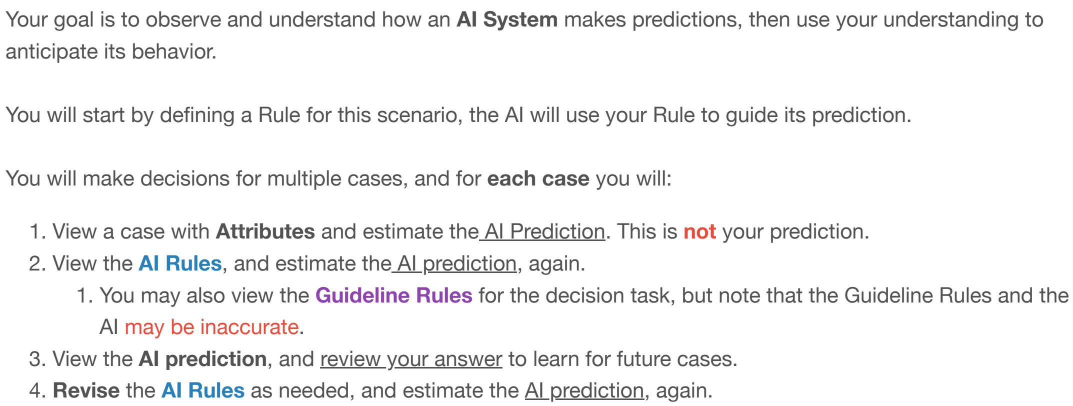
Based on the attributes above one way to make decisions is with the following **Guideline Rules**:

Note that the **AI Tool** *may* **use different rules** than the rules above to try to be more accurate for specific cases. But be careful that the AI may be wrong sometimes, so use **your judgement**.

Which of the following are True about the rules above (Multiple choice):

- □ All people older than 60 have high risk of heart disease.
- □ Resting blood pressure of 150 indicates this person have low risk of heart disease.
- □ A man of 60 years old, resting blood pressure 155, cholesterol level 60, is classified as high risk.
- □ Someone with a resting blood pressure of 160 is of high risk, no matter what the other attributes.

Figure A.15: Tutorial on the Heart Disease Prediction Task and screening questions to check the users' interpretation.



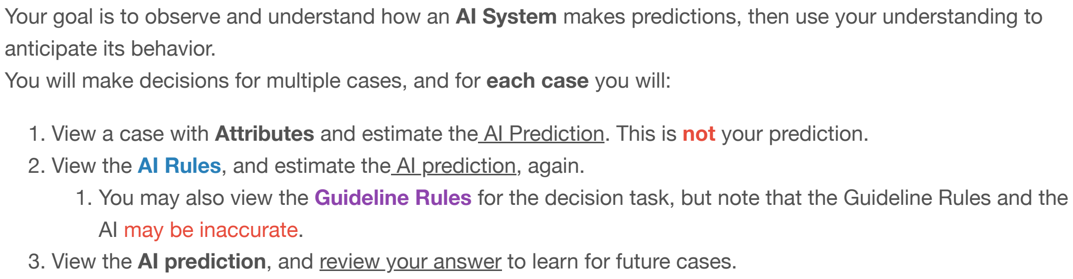
Your goal is to observe and understand how an **AI System** makes predictions, then use your understanding to anticipate its behavior.

You will start by defining a Rule for this scenario, the AI will use your Rule to guide its prediction.

You will make decisions for multiple cases, and for **each case** you will:

1. View a case with **Attributes** and estimate the AI Prediction. This is **not** your prediction.
2. View the **AI Rules**, and estimate the AI prediction, again.
    1. You may also view the **Guideline Rules** for the decision task, but note that the Guideline Rules and the AI may be inaccurate.
3. View the **AI prediction**, and review your answer to learn for future cases.
4. **Revise** the **AI Rules** as needed, and estimate the AI prediction, again.

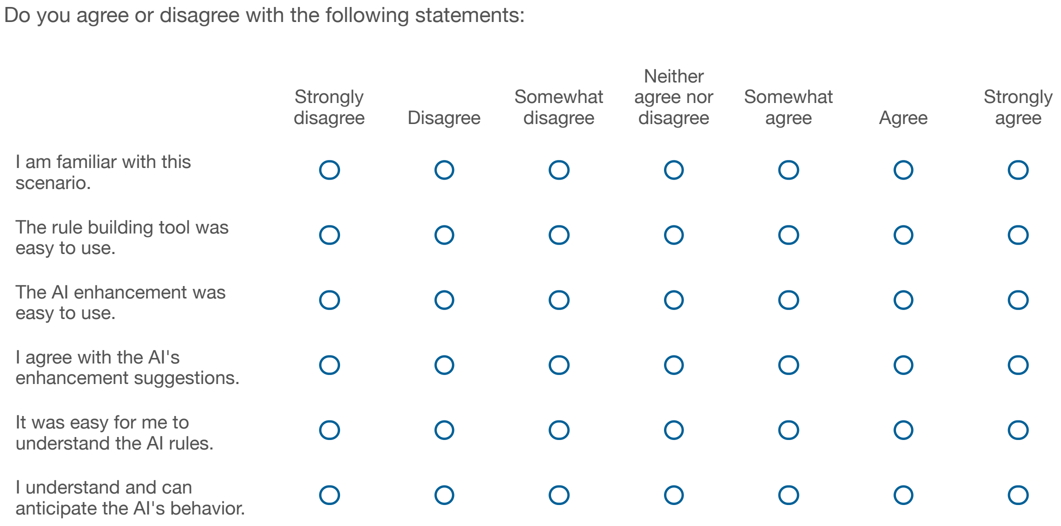
Figure A.16: Task description for Editable and CoExplain explanation.



Your goal is to observe and understand how an **AI System** makes predictions, then use your understanding to anticipate its behavior.
You will make decisions for multiple cases, and for **each case** you will:

1. View a case with **Attributes** and estimate the AI Prediction. This is **not** your prediction.
2. View the **AI Rules**, and estimate the AI prediction, again.
    1. You may also view the **Guideline Rules** for the decision task, but note that the Guideline Rules and the AI may be inaccurate.
3. View the **AI prediction**, and review your answer to learn for future cases.

Figure A.17: Task description for Read-only explanation.



Do you agree or disagree with the following statements:

	Strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
I am familiar with this scenario.	○	○	○	○	○	○	○
The rule building tool was easy to use.	○	○	○	○	○	○	○
The AI enhancement was easy to use.	○	○	○	○	○	○	○
I agree with the AI's enhancement suggestions.	○	○	○	○	○	○	○
It was easy for me to understand the AI rules.	○	○	○	○	○	○	○
I understand and can anticipate the AI's behavior.	○	○	○	○	○	○	○

Figure A.18: Likert Scale for perceived rating.