A Sim2Real Approach for Identifying Task-Relevant Properties in Interpretable Machine Learning

Eura Nofshin [*1] **Esther Brown** [*1] **Brian Lim** [2] **Weiwei Pan** [1] **Finale Doshi-Velez** [1]

Abstract

Existing user studies suggest that different tasks may require explanations with different properties. However, user studies are expensive. In this paper, we introduce XAIsim2real, a *generalizable, cost-effective* method for identifying task-relevant explanation properties in silico, which can guide the design of more expensive user studies. We use XAIsim2real to identify relevant properties for three example tasks and validate our simulation with real user studies.

1. Introduction

Recent literature suggests that explanations with different properties are useful for different tasks (e.g. Liao et al., 2022). For example, in an AI-auditing task, the user may need to check if the AI relied on a forbidden feature, such as gender, in computing a credit score (e.g. Kaur et al., 2020). In this case, we want explanations that are *faithful*; that is, they reliably capture the underlying behavior of the function. For a different task, to help a user quickly understand how a function produces its output, we may want explanations with *low complexity*, so the user can reason using the explanation in a limited amount of time (Poursabzi-Sangdeh et al., 2021).

Unfortunately, identifying which explanation properties are needed for what tasks remains an open challenge (e.g. Hase and Bansal, 2020). Prior works produce explanations with specific properties (e.g. Zhou et al., 2021), but do not test whether these properties improve human task-performance. Furthermore, many explanation properties, like "faithfulness", have multiple different formalizations in the literature (e.g. Chen et al., 2022b). Given the large number of properties, it is expensive and impractical to rely solely on user-studies for discovering task-relevant ones.

Drawing inspiration from the idea of sim2real used in other fields, such as robotics (see surveys Höfer et al., 2021; Kadian et al., 2020), in this work, we introduce XAIsim2real, a *generalizable, cost-effective* method for identifying task-relevant explanation properties in-silico that can guide the design of more expensive user studies. In XAIsim2real: (1) we choose a set of properties and optimize explanations for them, (2) we require details on learning and memory in our proxies for human task-performance (3) we use the proxy human to identify task-relevant properties. Finally, we validate insights from our computational pipeline via user-studies. By linking explanation properties directly to human task-performance, we hope to discover *generalizable* knowledge about what type of explanation is helpful for which task and why. Furthermore, our experiments show that our pipeline can serve as a useful precursor to real human-studies, by reducing a large hypothesis space.

Related Works

Several user-studies test which explanations (selected from a small set of possibilities) better aid humans in different tasks, such as predicting apartment prices (Poursabzi-Sangdeh et al., 2021; Lage et al., 2019a). In these studies, properties are carefully pre-selected based on domain knowledge. In contrast, XAIsim2real allows us to compare the effect of arbitrary explanation properties on the performance of human proxies *before* running user studies. Many works provide automatic evaluations of explanation quality through computing their properties (e.g. Liu et al., 2021; Bhatt et al., 2021; Nguyen and Martínez, 2020; Lukas and Garcia). But because the proposed property definitions are not grounded in tasks or evaluated on real human performance, there is a lack of consensus on how to formalize and apply these properties. In contrast, we propose a fully general approach for explicitly linking explanation properties to downstream task-performance. Thus, using our pipeline, we can identify which specific formalization of which property is useful for which task.

Human proxies have appeared in other areas of interpretable machine learning (Virgolin et al., 2020; Hilgard et al., 2021; Lage et al., 2018). Unlike us, these human

[*]Equal contribution [1]Department of Computer Science, Harvard University [2]Department of Computer Science, National University of Singapore. Correspondence to: Eura Shin <eurashin@g.harvard.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

proxies are not used to evaluate explanations. Similar to us, Chen et al. (2022a) simulate user studies with proxy humans. Unlike us, their simulations do not account for explanation properties. Thus, they do not provide reasons for *why* certain explanations, such as LIME, improved human performance. We focus on mapping of explanation properties to task-performance. Furthermore, we show by using a more transparent human proxy model instead of an arbitrarily flexible one (an MLP is used in Chen et al. (2022a)), we can check how our choice of human model affects the properties to task-performance mapping.

2. XAIsim2real: Connecting Explanation Properties to Human Performance

Our XAIsim2real consists of four components: (1) a set of tasks, (2) a set of explanation properties, (3) a computational proxy of the human that explicitly models memory and task learning, (4) a set of underlying ML functions being explained, and (5) a method for optimizing explanations to a given subset of properties. Below, in addition to defining each XAIsim2real component, we instantiate the component for our computational and user studies.

Notation. We assume an underlying *function* $\hat{y} = f(\mathbf{x})$. The *explanation method*, $E(f, \mathbf{x})$, provides the human with an explanation for function f local to the input $\mathbf{x}$. Though our pipeline is fully general, here, we focus on a *feature attributions*, which gives a weight for every input dimension. For a D-dimensional input $\mathbf{x}$, the explanation $E(f, \mathbf{x})$ is a D-dimensional weight vector. We use subscripts to denote explanations selected for a certain property, e.g. a faithful explanation is denoted E_{faithful}.

We focus on *local* tasks – the (real or proxy) human h make case-by-case decisions. We refer to the information available to the human as *the human input*, denoted $\mathbf{x}_h = [\mathbf{x},\ E(f, \mathbf{x}), \ldots]$. We denote human-produced outputs for the task as $y_h = h(\mathbf{x}_h)$. Note that the human's task y_h need not be the same as the function's output $\hat{y}$.

2.1. Component 1: Tasks.

Computational Instantiation. We consider two popular decision-making tasks from literature: forward prediction and forbidden features. Prior works indicate that they will likely require different properties: humans performing forward prediction prefer *sparse* (Poursabzi-Sangdeh et al., 2021; Hase and Bansal, 2020) and *faithful* explanations (Lertvitt and Toni, 2019), while humans performing forbidden features prefer *faithful* explanations (Liao et al., 2022).

Forward prediction: the human uses an explanation to predict a function's output. The *human's input*, $\mathbf{x}_h = [\mathbf{x}, E(f, \mathbf{x})]$, consists of the input $\mathbf{x}$ and explanation E. The *human's target* is the function output, $y_h^* = f(\mathbf{x})$.

Forbidden features: the human uses an explanation to determine if the function used a forbidden feature to compute its output. The *human's inputs*, $\mathbf{x}_h = [\mathbf{x}, E(f, \mathbf{x}), f(\mathbf{x})]$, consists of the input $\mathbf{x}$, explanation E, and function's output $f(\mathbf{x})$. The *human's target* is binary: did the function use a forbidden feature ($y_h = 1$) or no ($y_h = 0$). The correct human answer is $y_h^* = \mathbb{I}\{f(\mathbf{x}) = f(\mathbf{x} \text{ without } d)\}$, with d as the forbidden feature.

User Study Instantiation. In our user study, we modify a toy decision-making scenario related to medically treating aliens (Lage et al., 2019b; Swaroop et al., 2024). The toy scenario, rather than a more realistic one, mitigates confounding from different levels of task-specific prior knowledge. Furthermore, to mitigate confounding from different levels of algorithmic mistrust, we refer to explanations as information from an "alien researcher."

For *forward prediction*, users diagnose an alien with imaginary physical traits. The diagnosis is binary: healthy or not. The underlying function determines the true mapping of physical traits to a diagnosis. We provide explanations as an "alien researcher's" advice on how physical traits affect alien health. In the *forbidden features* task, users decide whether or not a doctor relied on a forbidden trait to diagnose aliens. Users see the "alien researcher's" opinion on traits used by the doctor. The underlying function determines whether the doctor truly used the trait for diagnosis. Screenshots of the UI are in Appendix C.1.

2.2. Component 2: Explanation Properties.

Computational Instantiation. We focus on three properties— robustness, faithfulness, and complexity— that are most relevant for feature attribution explanations. There are multiple formalizations for each property; we choose formalizations that are both commonly used and can be applied to any feature-based explanation.

Robustness measures the variation in a function's explanation when the input is perturbed. We use the local-stability formalization (Alvarez Melis and Jaakkola, 2018). *Faithfulness* evaluates how well the explanation matches the function's behavior. We use the local-infidelity formalization (Yeh et al., 2019). *Complexity* is a proxy for the cognitive burden for engaging with the explanation. We use a sparsity formalization (all equations in App A.1).

User Study Instantiation. Explanations shown to users are optimized for specific properties.

2.3. Component 3: Property Optimized Explanations.

Computational Instantiation. We optimize explanations for a given property (e.g. faithfulness or robustness) formalized as a mathematical loss function. In this paper, rather than solving this optimization algorithmically, we

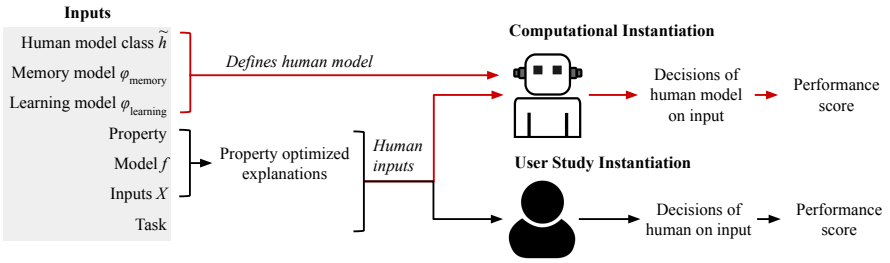


Figure 1: An overview of our XAIsim2real pipeline (robot, red arrows) compared to a user study (human, black arrows).

use our knowledge of the underlying functions to directly identify optimal explanations for different properties. For example, when the underlying function is linear around an input, we return, as our explanation, the weights of the linear function. This explanation will be optimal under any formalization of faithfulness. We consider four types of faithful explanations: neither sparse nor robust (E_{faithful}), sparse but not robust (E_{sparse}), robust but not sparse (E_{robust}), and both sparse and robust ($E_{\text{sparse+robust}}$).

User Study Instantiation. Users see the property-optimized explanations as advice from an "alien researcher." Users see the same (initially randomly assigned) type of explanation throughout the entire study.

2.4. Pipeline Component: ML Functions.

Computational Instantiation. We consider two underlying machine learning functions, f_{box} and f_{piece}, which we design to demonstrate the trade-offs between different explanation properties (details in Appendix A.2).

f_{box} (Figx 3a) is an inherently sparse binary classifier, using only a small subset of the total features for predictions. We can be certain that sparse explanations will provide reasonable insights on the function's computational process.

f_{piece} is a function for which different features are important for different inputs. Importantly, in f_{piece}, many weights are small but still non-zero. These small effects help us distinguish faithful explanations that include all (marginally) important features and sparse explanations that include only the most important ones.

User Study Instantiation. In the forward prediction task, f_{box} or f_{piece} determines the mapping from the alien's physical traits to a diagnosis. In the forbidden features task, they represent the doctor's decision-making process (i.e. whether the doctor uses the forbidden feature).

2.5. Pipeline Component: Proxy Human Model.

The proxy human model $\tilde{h}$ maps human inputs $\mathbf{x}_h$ to human outputs y_h. Specifying the proxy human model requires choices of *memory model* and *task learning model*. The *memory model* makes explicit our assumptions about how humans process inputs. We formalize this as a data *preproccessing* step. The *learning model* captures how humans learn a task based on experience. We formalize this as the process for optimizing $\tilde{h}$ on training data.

Computational Instantiation. We use a decision tree of up to depth two for $\tilde{h}$, as logic-based models are generally considered interpretable to humans (and therefore thought to mimic their decision-making) (Lage et al., 2019a).

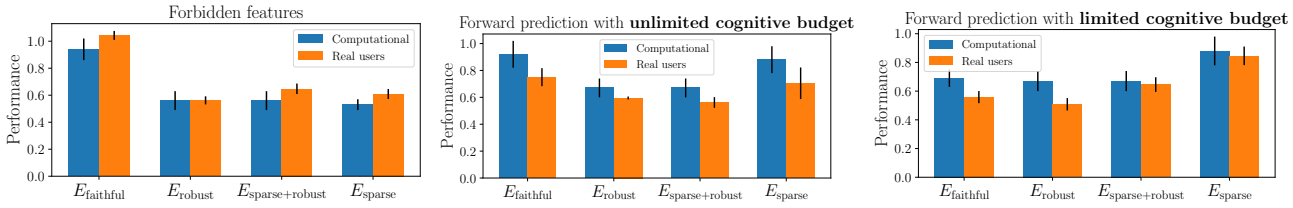
We consider two *memory models*. The first human model, $\tilde{h}_{\text{limited}}$, assumes a limited cognitive budget – the human can perform a limited number of mathematical operations. In h_{limited}, the human computes a partial inner product between $\mathbf{x}$ and $E(f, \mathbf{x})$, only using the two largest feature attribution weights. Our second model, $\tilde{h}_{\text{unlimited}}$, computes the full inner product. The inner product represents how the human combines the explanation and the input.

User Study Instantiation. We test both versions of our memory model. In one user study, we use time pressure to create conditions under which the human's cognitive budget is limited. Specifically, during the study, participants see both a global timer for how much time they have remaining to complete the questions and a local timer which counted down a "recommended" time per task – total time divided by total number of questions (Swaroop et al., 2024). In the other study, no time pressure is applied.

3. Experiments

The explanation property – denoted E_{faithful}, E_{sparse}, E_{robust}, or $E_{\text{sparse+robust}}$ – is our independent variable, and the human's task-performance is the dependent variable. The remaining pipeline components (tasks, functions, inputs) determine the experiment setting.

Computational experiments. We consider eight total settings combining two tasks (forward prediction, forbidden features), two functions (f_{box}, f_{piece}), and two human models with different cognitive budgets (h_{limited}, $h_{\text{unlimited}}$) (Section 2). Explanations are optimized on a set of 500 points near the function's decision boundary. Human models,



(a) (b) (c)

Figure 2: Different tasks require different properties (best explanation changes between plots), and XAIsim2real successfully proxies real human performance (orange follows blue). Bars are 95% confidence intervals.

however, are trained on a random subset of 10 of these points (we report the result of ten trials over these subsets). Test points are a set of 500 evenly spaced points.

Procedure for user studies. Our study is conducted via Prolific, an online crowdsourcing platform. We host each task as a separate study. Within a task, we use a between-subjects design, and users are randomized to different explanation types. We recruit 32 participants per task (8 participants per explanation type). All studies in our paper are approved by the Internal Review Board at Harvard University, protocol number IRB15-2076.

A participant experiences "train" and "test" phases in our user-study, which mirrors in-silico train and test. During *training*, users interact with ten practice questions for which correct decisions are given. They are instructed to form a decision-making strategy and can click through the practice problems until they feel prepared to proceed (there is no time pressure). During *testing*, participants answer thirty questions. Explanations in the test phase have the same properties as those shown in training. This process matches our simulation, in which each human model is trained and tested on one type of explanation.

3.1. Results

Using XAIsim2real, we reduce many hypotheses to a few promising candidates for user-studies. Our computational experiments test the human model's performance in eight possible settings. Three settings resulted in interesting hypotheses: (1) for forward prediction with f_{box} and h_{limited}, humans will perform best with E_{sparse}, (2) for forward prediction with f_{box} and $h_{\text{unlimited}}$, humans will perform equally well with E_{faithful} and E_{sparse}, (3) for forbidden features with f_{piece}, regardless of human model, humans will perform best with E_{faithful}.

We design user-studies for the above three settings, since, in each, the proxy human performed better with one explanation than the others (blue bars in Figure 2). In the remaining settings (full results in Appendix B.1), different explanation types did not yield different proxy human performance, eliminating the need to run user studies.

The performance of computational proxies transfers to humans. The goal of the human proxy is not to perfectly model human decision-making, but rather, to provide a sufficient proxy for ranking explanation properties (by task performance). In Figure 2, we provide evidence that our proxy model was sufficient; the ordering of explanation properties under our proxy humans (orange) matches the ordering under real humans (blue). Specifically, on forward prediction, E_{sparse} is best when there is time pressure (see Figure 2c), while E_{faith} does similarly well when the time pressure is removed (see Figure 2b). This confirms our hypothesis. As expected, in Figure 2a, E_{faith} is best for forbidden features. Overall, we expect (and see that) real humans perform slightly worse than their proxies.

4. Conclusion

In this paper, we introduce XAIsim2real, a sim2real approach for connecting explanation properties to human task-performance. We demonstrate XAIsim2real by addressing two research questions: (1) can we computationally link explanation properties to tasks? and (2) can we link the performance of computational proxies for humans to the performance of real humans? For (1), we use XAIsim2real to identify three tasks where we see, computationally, that different explanation properties are helpful for different tasks. For (2), we verify the property-task relationships we identify in-silico using user-studies.

This work is a proof-of-concept for an important research direction: efficiently identifying helpful explanation properties for human decision-making. Future work includes running studies on larger samples to measure statistical significance. The tasks, functions, and properties we used were hand-picked based on where we expected to see large effects; in future, we will explore more functions as well as algorithmic means to optimize explanations for properties. We will also explore human proxies other than a decision-tree, to test whether our results are robust to the choice of model.

5. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-2107391. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Umang Bhatt, Adrian Weller, and José MF Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3016–3022, 2021.

Valerie Chen, Nari Johnson, Nicholay Topin, Gregory Plumb, and Ameet Talwalkar. Use-case-grounded simulations for explanation evaluation. *Advances in neural information processing systems*, 35:1764–1775, 2022a.

Zixi Chen, Varshini Subhash, Marton Havasi, Weiwei Pan, and Finale Doshi-Velez. What makes a good explanation?: A harmonized view of properties of explanations. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022b.

Peter Hase and Mohit Bansal. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*, 2020.

Sophie Hilgard, Nir Rosenfeld, Mahzarin R Banaji, Jack Cao, and David Parkes. Learning representations by humans, for humans. In *International conference on machine learning*, pages 4227–4238. PMLR, 2021.

Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Sim2real in robotics and automation: Applications and challenges. *IEEE transactions on automation science and engineering*, 18(2):398–400, 2021.

Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. Human-in-the-loop interpretability prior. *Advances in neural information processing systems*, 31, 2018.

Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019a.

Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019b.

Piyawat Lertvitt and Francesca Toni. Human-grounded evaluations of explanation methods for text classification. *arXiv preprint arXiv:1908.11355*, 2019.

Q Vera Liao, Yunfeng Zhang, Ronny Luss, Finale Doshi-Velez, and Amit Dhurandhar. Connecting algorithmic research and usage contexts: a perspective of contextualized evaluation for explainable ai. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 10, pages 147–159, 2022.

Yang Liu, Sujay Khandagale, Colin White, and Willie Neiswanger. Synthetic benchmarks for scientific research in explainable machine learning. *arXiv preprint arXiv:2106.12543*, 2021.

Ondrej Lukas and Sebastian Garcia. Bridging the explanation gap in ai security: A task-driven approach to xai methods evaluation.

An-phi Nguyen and María Rodríguez Martínez. On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*, 2020.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–52, 2021.

Siddharth Swaroop, Zana Buçinca, Krzysztof Z Gajos, and Finale Doshi-Velez. Accuracy-time tradeoffs in ai-assisted decision making under time pressure. In *29th International Conference on Intelligent User Interfaces (IUI'24). ACM*, 2024.

Marco Virgolin, Andrea De Lorenzo, Eric Medvet, and Francesca Randone. Learning a formula of interpretability to learn interpretable formulas. In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II 16*, pages 79–93. Springer, 2020.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in neural information processing systems*, 32, 2019.

Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593, 2021.

A. Details of Computational Instantiations

A.1. Definitions of Explanation Properties

Robustness measures the variation in a function's explanation when small changes are made to the input. We consider a specific formalization called *local stability* (Alvarez Melis and Jaakkola, 2018), which encapsulates many other formalizations of robustness as specializations (Chen et al., 2022b).

$$\text{local-stability}(E, f, \mathbf{x}, r) = \max_{\|\mathbf{x}-\mathbf{x}'\| \leq r} \frac{\|E(f, \mathbf{x}) - E(f, \mathbf{x}')\|}{\|\mathbf{x} - \mathbf{x}'\|}. \tag{1}$$

The hyperparameter r defines the "locality" of the definition; the bigger the r, the more globally unchanging the explanation must be to evaluate well.

Faithfulness evaluates how well the explanation matches the function's true behavior. We consider a specific formalization called *Local infidelity* (Yeh et al., 2019), which has been applied in literature to analyze a number of explanation methods (Chen et al., 2022b). Local infidelity evaluates how well the explanation method E can reproduce the behavior of the function f "locally," near an input $\mathbf{x}$, instead of over the entire dataset:

$$\text{local-infidelity}(E, f, \mathbf{x}, p) = \mathbf{E}_{p(\mathbf{x}'|\mathbf{x})} \left[L \left(f(\mathbf{x}), g(\mathbf{x}', E(f, \mathbf{x})) \right) \right], \tag{2}$$

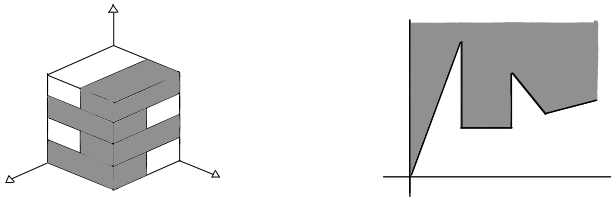
where $\mathbf{E}$ is the expectation, and p is a distribution over points centered on $\mathbf{x}$, and L is any loss function appropriate for the task.

Complexity measures the intricacy of the explanation and is a proxy for the cognitive burden on the human when engaging with the explanation. We measure the complexity of a feature attribution explanation using *sparsity*, which counts the number of non-zero features:

$$\text{sparsity}(E) = \sum_{d}^{D} \mathbb{I}(E(f, \mathbf{x})_d \neq 0) \tag{3}$$

Here, D is the number of features, $E(f, \mathbf{x}_d)$ refers to the attribution of the d-th feature, and $\mathbb{I}$ is an indicator function.

A.2. Definitions of functions



(a) f_{box} function (b) f_{piece} function (2D example)

Figure 3: Visualization of the two underlying machine learning classification functions we used in our experimental setting. Different colors are different labels.

Model f_{box}. This function takes in three input dimensions. At any given input, this function relies on only one input feature to make a decision. Depending on the input region, the function switches between using one of two input features, $\mathbf{x}_1$ or $\mathbf{x}_2$, to generate its output. The switch is determined by a third input feature, $\mathbf{x}_3$. Formally, the *Box function* is defined as follows:

$$f_{\text{box}}(\mathbf{x}) = \begin{cases} \mathbb{I}\{\mathbf{x}_2 > 0.5\}, & \mathbf{x}_3 \leq 0.25 \\ \mathbb{I}\{\mathbf{x}_1 > 0.5\}, & 0.25 < \mathbf{x}_3 \leq 0.5 \\ \mathbb{I}\{\mathbf{x}_2 > 0.5\}, & 0.5 < \mathbf{x}_3 \leq 0.75 \\ \mathbb{I}\{\mathbf{x}_1 > 0.5\}, & 0.75 < \mathbf{x}_3 \end{cases} \tag{4}$$

where $\mathbb{I}(\text{condition}(x))$ is the indicator function that returns 1 if $\text{condition}(x)$ is true, 0 otherwise. In Eq. 4, the binary classification depends on either the first $\mathbf{x}_1$ or second $\mathbf{x}_2$ input feature. The third feature $\mathbf{x}_3$ decides which of the two is

used. For example, the first condition of f_{box}'s definition says, if $\mathbf{x}_3$ is less than 0.25, then the classification returns 1 if $\mathbf{x}_1$ is greater than 0.5 and 0 otherwise.

Model f_{piece}. We define a piecewise linear function with 4 pieces and 10-dimensional inputs as follows:

$$f_{\text{piece}}(x) = \begin{cases} \mathbb{I}\left\{\mathbf{x}^\top W_{1:} > 0\right\}, & \mathbf{x}_1 \leq 0.25 \\ \mathbb{I}\left\{\mathbf{x}^\top W_{2:} > 0\right\}, & 0.25 < \mathbf{x}_1 \leq 0.5 \\ \mathbb{I}\left\{\mathbf{x}^\top W_{3:} > 0\right\}, & 0.5 < \mathbf{x}_1 \leq 0.75 \\ \mathbb{I}\left\{\mathbf{x}^\top W_{4:} > 0\right\}, & 0.75 < \mathbf{x}_1, \end{cases} \tag{5}$$

where $W_{i:}$ refers to the i-th row of the weight matrix, defined as

$$W = \begin{pmatrix} 0 & 1 & -1 & 0 & 1 & -0.1 & 0.1 & -0.1 & 0.1 & -0.1 & -0.7 \\ 0 & -0.8 & -0.2 & 0.2 & 0.1 & -0.9 & -0.1 & -0.1 & 0.1 & -0.2 & 1 \\ 0 & -0.8 & -0.2 & 0 & 0.1 & -0.9 & -0.1 & -0.1 & 0.1 & -0.2 & 1 \\ 0 & -0.05 & 1 & -0.8 & -0.1 & 0.1 & 0.9 & -0.2 & 0.1 & 0.8 & -1 \end{pmatrix}$$

B. Full results

B.1. Computational results for all eight settings

Tables 1 and 2 report the human model's performance on our tasks. In these tables, we find evidence supporting our hypotheses. Specifically, in Table 1, E_{faith} is important for the forward prediction task when the human model has no constraints on the cognitive budget ($\tilde{h}_{\text{unlimited}}$). When constraints are introduced ($\tilde{h}_{\text{limited}}$), E_{sparse} remain the sole important feature for f_{box}. Note E_{sparse} is not helpful for f_{piece} because the function depends densely on all of the inputs.

Finally, for the forbidden features task in Table 2, E_{faith} is the most important property for explaining on f_{piece}, as expected. Performances on the forbidden features task for f_{box} are all perfect because we define the forbidden feature to be one that is always used by f_{box} (and therefore, there was only one label). This is a scenario in which the explanation did not help the human model with the task, and thus we expected no difference in performance among the explanation types; this is indeed the case. This result is an example where the sim2real pipeline informs us that that variation in explanation properties is unlikely to impact real human performance on the forbidden features task using f_{box} as the underlying function. Thus, this would not be a setting we would test in our user studies.

		E_{faithful}	E_{robust}	E_{sparse}	$E_{\text{sparse+rob.}}$
f_{box}	$\tilde{h}_{\text{limited}}$	0.69 ± 0.06	0.67 ± 0.07	$\mathbf{0.88 \pm 0.1}$	0.67 ± 0.07
	$\tilde{h}_{\text{unlimited}}$	$\mathbf{0.92 \pm 0.1}$	0.67 ± 0.07	$\mathbf{0.88 \pm 0.1}$	0.67 ± 0.07
f_{piece}	$\tilde{h}_{\text{limited}}$	0.51 ± 0.04	0.54 ± 0.04	0.51 ± 0.04	0.54 ± 0.04
	$\tilde{h}_{\text{unlimited}}$	$\mathbf{0.99 \pm 0.0}$	0.54 ± 0.04	0.51 ± 0.04	0.54 ± 0.04

Table 1: Performance of proxy human on prediction with 95% confidence intervals. Rows are each property optimized explanation. Higher is better; performance is over 10 trials.

		E_{faithful}	E_{robust}	E_{sparse}	$E_{\text{sparse+rob.}}$
f_{box}	$\tilde{h}_{\text{limited}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	$\tilde{h}_{\text{unlimited}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
f_{piece}	$\tilde{h}_{\text{limited}}$	$\mathbf{0.94 \pm 0.08}$	0.56 ± 0.07	0.53 ± 0.04	0.56 ± 0.07
	$\tilde{h}_{\text{unlimited}}$	$\mathbf{0.94 \pm 0.08}$	0.56 ± 0.07	0.53 ± 0.04	0.56 ± 0.07

Table 2: Performance of proxy human on forbidden features with 95% confidence intervals. Rows are each property optimized explanation. Higher is better; performance is over 10 trials.

B.2. Results of Optimizing Explanations to Properties

In Table 3, we verify that each explanation scores best on the property for which it was optimized. For example, E_{robust} and $E_{\text{sparse+robust}}$ score best on both robustness metrics (max sensitivity and local robustness) when compared to all other explanation types. Second, explanations score similarly on properties for which they are not being optimized. For example, all of the non-sparse explanations (E_{faith} and E_{robust}) evaluate to similar levels of sparsity.

Model	Property	E_{faithful}	E_{robust}	E_{sparse}	$E_{\text{sparse+rob.}}$
f_{box}	Local Infidelity	$\mathbf{0.01 \pm 0.0}$	0.34 ± 0.03	$\mathbf{0.04 \pm 0.01}$	0.36 ± 0.03
	Sparsity	$\mathbf{3.91 \pm 0.02}$	4.00 ± 0.00	$\mathbf{2.0 \pm 0.0}$	$\mathbf{2.0 \pm 0.0}$
	Local Stability ($r = 0.1$)	46.34 ± 17.37	$\mathbf{0.0 \pm 0.0}$	59.83 ± 2.85	$\mathbf{0.0 \pm 0.0}$
f_{piece}	Sparsity	9.46 ± 0.03	$\mathbf{9.0 \pm 0.0}$	$\mathbf{2.0 \pm 0.0}$	$\mathbf{2.0 \pm 0.0}$
	Local Infidelity	$\mathbf{0.0 \pm 0.0}$	0.42 ± 0.03	0.45 ± 0.03	0.42 ± 0.03
	Local stability ($r = 2$)	15.18 ± 0.30	$\mathbf{0.0 \pm 0.0}$	$\mathbf{13.09 \pm 0.25}$	$\mathbf{0.0 \pm 0.0}$

Table 3: Optimization results. Numbers are mean property values of the explanation at each input, with 95% confidence intervals. Rows are each property optimized explanation. Lower is more optimized. Bolded numbers are outside the CI of unbolded numbers. For local stability, r refers to the radius parameter from Eq. 1. For local infidelity, p refers to the distribution centered on the input point from Eq. 2.

Some values in Table 3 are zero due to our process for optimizing the explanations. The robust explanations produce a local infidelity score of zero because, to optimize for robustness, we choose a global explanation that is as faithful as possible. Since these explanations are global, they do not change with the inputs and evaluate to zero under the definition of local stability in Eq. 1. This also means the sparsity level of each robust explanation is constant and results in zero variance. Similarly, the sparse explanations have zero variance because, to optimize for sparsity, we always pick the two largest feature attributions (the sparsity level is always two). Finally, the faithful explanations are zero because, to optimize for faithfulness, we return the ground-truth weights of the underlying function. Since these weights are the same as the function, the loss will be zero when evaluated at each point. The non-zero values (Local Infidelity is 0.01 for E_{faithful} on f_{box}) are caused by the fact that we round the optimized weights in our human model.

C. User Study Details

The main study has four parts: (1) instructions; (2) comprehension check questions; (3) a task training phase; and (4) the main study testing phase. *Instructions* introduces the scenario and describes each UI element. *Comprehension checks* assess whether the participant understands the task well enough to form a proper strategy. For example, for forward prediction, we ask "According to the [alien] researcher, which measurement would have the *biggest effect* on the alien's health?", to check that participants know that a higher absolute value means a higher feature attribution. To mirror training and evaluation in-silico, we include "train" and "test" phases in our user-study. During *training*, users interact with ten practice questions for which correct decisions are given. They are instructed to form a decision-making strategy and can click through the practice problems until they feel prepared to proceed (there is no time pressure during training). During *testing*, participants answer thirty questions. Explanations in the test phase have the same properties as the explanations shown in training. This process matches our simulation, in which each human model is trained and tested on one type of explanation.

C.1. User Interfaces

In Figure 4, we provide an example of the test phase interface the forward prediction and forbidden features tasks.

In Figure 4a, the first block describes the alien's measurements, corresponding to the inputs $\mathbf{x}$. The second block contains the explanations $E(f, \mathbf{x})$. The third block lets the user provide their decision on the diagnosis, which corresponds to $h(\mathbf{x}_h)$.

In Figure 4b, the first block denotes the forbidden feature. The second block describes the alien's measurements, corresponding to the inputs $\mathbf{x}$. The third block is the doctor's diagnosis for the alien, corresponding to $\hat{y}(\mathbf{x})$. The fourth block contains the explanations $E(f, \mathbf{x})$. The fifth block lets the user provide their decision on whether the forbidden feature was used in diagnosis, which corresponds to $h(\mathbf{x}_h)$.

C.2. Design Choices in Translating the Simulation Pipeline to Real Users

Some elements of the simulation pipeline are not straightforward to implement on real humans. In this section, we describe design choices made to align our simulation and real user studies.

Constructing training and test phases. Mirroring the training and evaluation phase in the computational pipeline, we also include "train" and "test" phases in our user-study. During the *training phase*, users interact with ten practice questions for which the correct decisions are given. They are instructed to form a decision-making strategy and can click through these practice problems until they feel prepared to proceed. These practice problems parallel the ten training points used to train our decision tree model $\tilde{h}$. There is no time limit or time pressure during this phase.

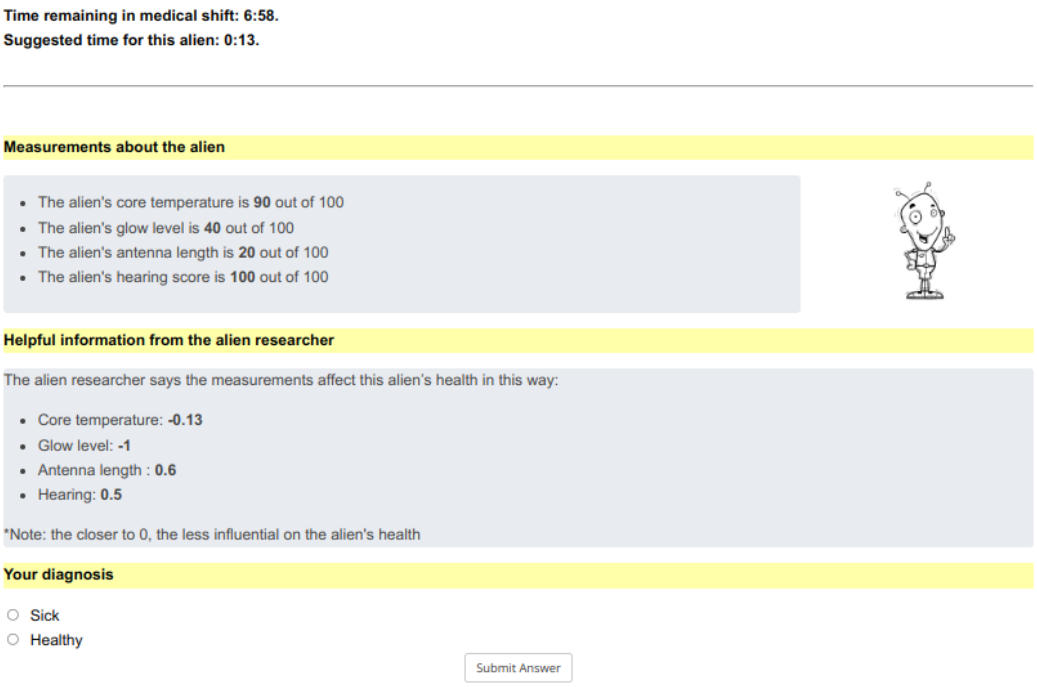
During the *testing phase*, participants are asked to provide decisions on thirty questions for which the correct answer is not given. The explanations during the test phase have the same properties as the explanations during the training phase. This process closely matches our simulation pipeline, in which each human model is trained and tested on one type of explanation.

Selection criteria for training points. Because we are limited in the number of examples we could provide, we select them to be as informative as possible. For example with forward prediction, we find in our pilot studies that users learn best from input points that are a mix of easy examples clearly belong to one class because they are far from the function's decision boundary and harder examples close to the function's decision boundary. This is opposed to the computational model $\tilde{h}$, which was trained only on points near the boundary. Note that we can use the decision boundary because we have full knowledge of the underlying functions, f_{box} and f_{piece}; in situations where the underlying function is not known, one could use the sim2real pipeline to identify a subset of training points that led to highest performance for the synthetic human and use these points for the user study.

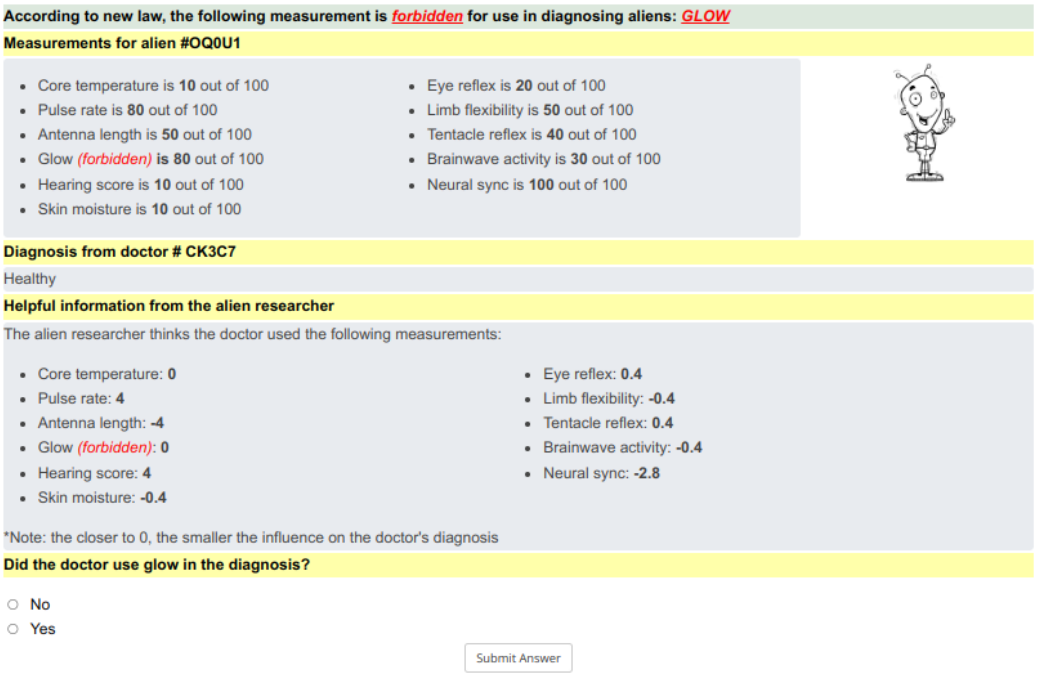
How we picked the thirty test points. We choose the test questions under a mixture of conditions that allow us to test both Q1 ("linking properties to tasks") and Q2 ("do simulation results match real human behavior"). Specifically, we choose ten test questions from each of the following categories:

- Questions for which the human proxy model performed the same, regardless of whether the explanation was robust, sparse, or faithful.
- Questions for which the human proxy model suggested that real users will perform *better* with the best explanation type (according to simulation) than a different explanation type.
- Questions for which the human proxy model suggested that real users will perform *worse* with the best explanation type (according to simulation) than with a different explanation type.

The best explanation types refer to the explanations that performed best on the task in the simulation experiments. For example, the "best" explanation type in the forbidden features task was the faithful explanations. Overall, this collection of test cases allows us to check whether the real user's behavior matched the human proxy model. Moreover, by testing these distinct cases in which we believe that we are giving the real users a combination of useful and less useful explanations, we can observe the real connection between tasks and properties.



(a) UI for forward prediction



(b) UI for forbidden features

Figure 4: Example of user interface for the study, test phase, on the forward prediction task of diagnosing aliens. During the training phase, participants could view a number of these examples, for as long as they wished, with the correct answers given. There was no timer during the training phase.