Mental Models of Mere Mortals with Explanations of Reinforcement Learning

ANDREW ANDERSON, JONATHAN DODGE, AMRITA SADARANGANI, ZOE JUOZAPAITIS, EVAN NEWMAN, JED IRVINE, SOUTI CHATTOPADHYAY, MATTHEW OLSON, ALAN FERN, and MARGARET BURNETT, Oregon State University

How should reinforcement learning (RL) agents explain themselves to humans not trained in AI? To gain insights into this question, we conducted a 124-participant, four-treatment experiment to compare participants' mental models of an RL agent in the context of a simple Real-Time Strategy (RTS) game. The four treatments isolated two types of explanations vs. neither vs. both together. The two types of explanations were as follows: (1) saliency maps (an "Input Intelligibility Type" that explains the AI's focus of attention) and (2) reward-decomposition bars (an "Output Intelligibility Type" that explains the AI's predictions of future types of rewards). Our results show that a combined explanation that included saliency and reward bars was needed to achieve a statistically significant difference in participants' mental model scores over the no-explanation treatment. However, this combined explanation was far from a panacea: It exacted disproportionately high cognitive loads from the participants who received the combined explanation. Further, in some situations, participants who saw both explanations predicted the agent's next action *worse* than all other treatments' participants.

CCS Concepts: • **Computer systems organization → User studies**; • **Computing methodologies →** *Inteligent agents*;

Additional Key Words and Phrases: Intelligent user interfaces, human-computer interaction

ACM Reference format:
Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Matthew Olson, Alan Fern, and Margaret Burnett. 2020. Mental Models of Mere Mortals with Explanations of Reinforcement Learning. *ACM Trans. Interact. Intell. Syst.* 10, 2, Article 15 (May 2020), 37 pages.
https://doi.org/10.1145/3366485

This work was supported by DARPA #N66001-17-2-4030. Any opinions, findings and conclusions or recommendations expressed are the authors' and do not necessarily reflect the views of DARPA, the Army Research Office, or the US government.
The reviewing of this article was managed by associate editor Chang, Remco.
Authors' address: A. Anderson, J. Dodge, A. Sadarangani, Z. Juozapaitis, E. Newman, J. Irvine, S. Chattopadhyay, M. Olson, A. Fern, and M. Burnett, School of EECS, Oregon State University, 1500 SW Jefferson Way, Corvallis, OR, 97331; emails: {anderan2, dodgej, sadarana, juozapaz, newmanev, irvine, chattops, olsomatt, Alan.Fern, burnett}@eecs.oregonstate.edu.

2160-6455/2020/05-ART15 $15.00
https://doi.org/10.1145/3366485

1 INTRODUCTION

Although eXplainable Artificial Intelligence (XAI) has seen increasing interest as AI becomes more pervasive in society, the challenges of understanding AI systems grow in both scope and import [57]. In this article, we focus on the human aspects of these challenges—how people form mental models, given different types of XAI explanations, and the cognitive loads they incur in forming these mental models.

People's mental models, in the context of XAI, are basically their understanding of the way the AI (the "agent" performing the action) works. More formally, mental models are "*internal representations that people build based on their experiences in the real world*" [38]. People's mental models vary in complexity and accuracy, but a good mental model will enable a person to *understand* system behavior, and a very good one will enable them to *predict* future behaviors.

Not all explanation types will lead people's mental model formation in the right directions. Further, consuming an explanation is not without cost. Measuring those costs can inform designers of the benefits of different explanation types versus the costs.

Recent work by Kulesza et al. introduced four principles for explaining AI systems to people who are not AI experts [25]. These principles were as follows: Be iterative, be sound, be complete, and do not overwhelm the user, where here the notions of soundness and completeness are analogous to "the whole truth (completeness) and nothing but the truth (soundness)." Kulesza et al.'s empirical results showed that explanations adhering to these principles enabled non-AI experts to build higher-fidelity mental models of the agent than non-AI experts who received less sound/complete explanations [25].

Using Kulesza et al.'s principles, we created two types of explanation, and in this article we investigate how people's mental models of a reinforcement-learning agent vary in response to these explanations. The explanation styles we investigated were saliency maps showing where the agent is "looking" and decomposed reward bars showing the agent's current prediction of its future score. In Lim and Dey's categorization of explanation types, saliency maps would fit into the "Input" category (information that goes "into" the AI's reasoning), and decomposed reward bars would fit into the "Output" category (the result of the AI's reasoning process that lead to its choice) [27, 28].

To do so, we conducted a controlled lab study with 124 participants across four treatments (saliency, rewards, both, neither) and measured their understanding of the agent and their ability to predict its decisions. We also gathered information using the NASA Task-Load indeX (TLX) [12], a validated survey that measures six dimensions of cognitive load, which allowed us the ability to measure the participants' cost for consuming the explanations.

Our investigation was in the context of Real-Time Strategy (RTS) games. However, publicly available RTS games have stringent time constraints, complex concepts, and myriad decisions, which would have introduced too many confounding variables for a controlled study. For example, we needed each participant to consider the *same* set of decisions. Thus, we built our own game, inspired by RTS, which we describe later.

In this context, we structured our investigation around the following research questions:

- **RQ-Describe** - Which treatment is better (and how) at enabling people to *describe* how the system works?
  - What cognitive loads do these explanations place on people?
- **RQ-Predict** - Which treatment is better (and how) at enabling people to *predict* what the system will do?
- **RQ-Predict-How** - What information about the agent do people draw upon to make the predictions they make?

2 BACKGROUND AND RELATED WORK

Since our investigation evaluates XAI approaches' effects on humans' mental models of an RL agent, three bodies of work are particularly relevant: work on humans' mental models of AI, work on RL interacting with humans, and work on XAI in domains like ours.

2.1 Mental Models and XAI

One criticism of XAI research has been that "*...most work... uses only the researchers' intuition of what constitutes a 'good' explanation*" [33]. Miller argues that XAI researchers need to build on the expansive knowledge in philosophy, psychology, and cognitive science, which has long studied human explanation evaluation, selection, presentation, and so on. Hoffman et al. operationalizes these goals by proposing a number of metrics of explanation quality, describing "goodness" (a priori expert judgment of the explanation's quality) and "satisfaction" (*a posteriori* judgment whether the explanation consumer finds it adequate in context) [14]. Unfortunately, "satisfaction" is a complex multifaceted construct, including, but not limited to "understandability, feeling of satisfaction, sufficiency of detail, completeness, usefulness, accuracy, and trustworthiness" [14]. Our study draws upon some of Hoffman et al.'s metrics, specifically understandability (i.e., how well can they describe it) and usefulness (i.e., how well can they predict an action).

Hoffman et al. use these definitions to propose that "*by hypothesis, explanations that are good and are satisfying to users enable users to develop a good mental model*" [14]. Hoffman et al. also points out that "*there is a consensus that mental models can be inferred from empirical evidence.*" One implication of the combination of these two remarks is that if participants find more misbehaviors of an AI agent with an explanation than without an explanation, then the researcher can conclude that the explanation helped their mental models. Our investigation subscribes to this premise and further investigates what was actually in the mental models our explanations facilitated and what aspects of the explanations participants drew upon to form them.

2.1.1 Measuring Mental Models in XAI. Hoffman et al. enumerate many elicitation methods for mental models (see Reference [14], Tables 4 and 5) and recommend using more than one method [14]. In this article, we focus on the "Prediction Task" [34] and "Task Reflection" [30] mental model elicitation methods. In accordance with their recommendations, we included free response questions with each prediction task (without a confidence score, due to time constraints) and analyzed responses qualitatively, as we describe in later sections.[1]

Previous work has found that providing explanations of AI generally improves mental models of the AI. For example, Tullio et al. examined participants' mental models for a system that predicted how "interruptible" their managers were over a six-week period, finding that explanation allowed participants to dispense with some misconceptions, but the overall structure of their mental model remained stable [51]. Bostandjiev et al. found that explanations led to an increase in satisfaction in a music recommendation system [4]. In the robotics domain, Wortham et al. found that a graph-based visualization of the agent's drives improved participants' mental models of the robot [60]. Lim et al.'s work focused on "intelligibility types" of information, showing that their participants demanded "why" information when the system behaved unexpectedly [28]. Kulesza et al. looked at the impact of two explanation properties, *completeness* and *soundness*, on their participants' mental models in a music recommendation system powered by a hybrid AI agent [26]. From their work, soundness is "the extent to which *each component of an explanation's content* is truthful

[1]Methods like these allow collecting very rich data, but the associated cost of the qualitative analysis needed to process such data are substantial. For those seeking a more lightweight design aid, see the conceptual framework provided by Wang et al. [56].

in describing the underlying system." Similarly, completeness is "the extent to which *all of the underlying system* is described by the explanation" [26]. They found that participants tended to do better at understanding the explanations of the AIs' features rather than of the AI's process and further that the participants' understanding of features increased as completeness increased. Their results helped inform the four principles behind their "explanatory debugging" approach, which they empirically investigated with a Multinomial Naive Bayes classifier [25]. Using two of Lim et al.'s intelligibility types ("inputs" and "outputs"), we build from Kulesza's four principles and apply them to explaining Reinforcement Learning (RL) agents.

2.2 Cognitive Load and Problem-Solving

When trying to solve problems (in this case, figure out an AI with the help of explanations), people can experience at least three types of cognitive load: intrinsic, germane, and extraneous [20, 24, 49, 53]. *Intrinsic* cognitive load is the inherent effort of figuring out the problem itself. *Germane* cognitive load is the effort of learning the explanation's content and applying it. We view these two types of cognitive load as valuable, or at least necessary.

The third type, *extraneous* cognitive load, is the effort unrelated to the problem at hand, such as hunting for missing information, figuring out which part of the explanation is the most relevant, or trying to figure out what the explanation is trying to communicate. To try to maximize people's learning or problem-solving success, researchers aim to minimize the extraneous load [20, 24, 49, 53], because if someone's finite energy is being expended on extraneous cognitive load, then they have less to apply to the problem at hand (i.e., their intrinsic/germane cognitive load).

There are many ways, both used and proposed, to measure participants' cognitive load, such as Zagermann's proposal to use eye-tracking data to measure cognitive load in visual computing [61]. However, gathering eye-tracking data for 124 participants would have been a costly undertaking, greatly slowing down how many participants we could run. There are other measures for perceived cognitive load, such as the Rating Scale of Mental Effort (RSME) by Zijlstra and Van Doorn [64], which is a single numeric dimension for participants to rate how much effort they felt that they spent [58]. However, a single scale does not allow us to separate the different mental tradeoffs that participants might make.

To gain insights into these tradeoffs, validated surveys like the NASA Task-Load indeX (TLX) are useful. The TLX was designed to measure the cognitive load that users experienced when interacting with human-machine interfaces [35]. It measures cognitive load across six dimensions: mental demand, physical demand, temporal demand, effort, success, and frustration, which allows researchers to understand the tradeoffs that participants experience The TLX has been used by a variety of researchers around the world, even 20 years after its creation, speaking to its effectiveness as a tool [11].

2.3 RL and Humans

One trend in RL research investigates humans' capability to speed up RL agent training. Thomaz et al.'s work focused on how humans teach an RL agent in a virtual domain, finding that users' agent-training strategies adjusted as their mental models adjusted, too [50]. Peng et al. and Rosenfeld et al. studied humans-in-the-loop with RL agents, from non-experts to RL experts, finding that humans can help speed up the learning process for RL agents by shaping the training signal [40, 43]. In contrast with these works, our work does not look at how humans can help *machines* but rather how machines can give *humans* what they need through evaluating explanations.

Other work focuses on human interpretability of RL agents. An example of the interpretability category is Huang et al.'s work on showing possible trajectories that their agent could take in the self-driving car domain, with the problem that with a lot of states, decision are non-critical (e.g.,

highway with no vehicles nearby), but there are also states that decisions are far more important (e.g., that same highway, heavy traffic [17]). Their "explanation" was a visualization of four possible actions (such as "Merge Behind," "Slow Down," etc.) that their agent would possibly take. They asked participants to select the action that matched their mental model of how the agent behaved, a form of prediction task. Their results compared the amount of trust granted by participants to two agents with varying amounts of training. They derived a function to identify "critical" states in the self-driving car domain and found that presenting the critical states to participants allowed them to allocate more trust to the agent with more training. Our work allowed us to show the participants every state (only 14) and assess the capability of either explanation on their ability to predict actions. Greydanus et al. used a perturbation-based saliency method to evaluate how well AI non-experts were able to detect overfit RL policies, and their results pointed at the usefulness of saliency maps when the agent was looking at the "wrong thing," such as the hint pixels they added to the frames of the game (see their Figure 5(f)) [10]. Their results indicated that saliency maps at each individual decision could assist with identifying overfit policies in RL agents at the end of a game, manifest by *only* focusing on hint pixels and ignoring the rest of the domain. Our contributions differ from theirs, because their work focused on *selecting* an agent, whereas our work focuses on *predicting* actions and *describing* the agent's algorithm.

One of the challenges of explaining RL systems is whether to explain an individual decision or the whole policy. Some recent work focuses on global explanations of the agent's policy [2, 52]. For example, van der Waa et al.'s policy explanation states what the agent will do for the next n actions and what situations it expects it will encounter, rather than a local, more focused explanation pertaining to the current state [52].

However, Hohman et al. subscribe to a different approach: "...multiple explanations are often used to gain an ultimate interpretation of a model" [15]; i.e., that the sum of local explanations should lead to an understanding of the global policy. This is the approach used in our investigation, in which we explain each decision locally, and then elicit a global "policy explanation" from participants at the end (i.e., **RQ-Describe**). Similarly, another body of work focuses on explaining individual decisions [10, 31, 60] or fragments of the policy (e.g., "What will you do when a human is near you") [13]. Wortham et al. explained the current instantaneous time-slice, whereas Hayes' explained the agent's policy in a more global sense via responding to queries (e.g., "Why didn't you inspect the part?"). To answer these queries, they applied a Boolean algebra to a set of predicates that adequately described the states where the desired action would be taken. Wortham et al. focused on improving agent transparency through passive engagement with a explanation, where their measurements focus on the participants' perceptual response to the passive display of the agent's current action (e.g., Is the robot thinking? Describe robot task?) [60]. Our work differs, because we actively measure the participants' understanding of the agent's decision process by their ability to predict its next action, as opposed to whether the participant perceives that they understand it (e.g., "Understand objective? (Y/N)" [60]). Further, Hayes et al.'s explanations are natural-language based, whereas our own explanations allow participants to interpret data and interactively develop their own interpretation (i.e., "self-explanation" [5]).

One of the most similar works to our own used local explanations and measured their efficacy via a prediction task, satisfaction, and trust [32] using StarCraft 2 (a popular Real-Time Strategy game) as a domain. They found that their explanation approach performed significantly better than baselines. Our work differs by evaluating the impact of combinations of multiple explanations on participants' mental models, and we gather two mental model metrics. Though their work has a similar number of participants (120), they deployed their study as an online survey, whereas our work was performed in the lab, allowing for greater control over variation between participants. Our prediction task differs from theirs in a couple of ways: (1) In our domain, the participants

could predict anything that was in the potential action space, whereas they provided a multiple choice question over a subset of potential actions that their agent could have taken at a decision point. (2) Their work did not report on any qualitative data. Our work aims to fill these gaps (e.g., participants justifying *why* they made their predictions).

2.4 XAI and Real-Time Strategy Domains

AI researchers have been using Real-Time Strategy (RTS) games such as StarCraft for some time [39], with some recent results showing that RL agents can beat human experts [55]. Kim et al. showed in StarCraft that different people across a range of skill levels in the domain preferred different aspects of an AI, such as micro-management and decision making [22, 23]. Dodge et al. investigated what explanations might look like for RTS by investigating how expert explainers provided commentary on games of StarCraft 2, capturing explanation *supply* [7]. Penney et al. then looked at how RTS domain experts tried to make sense of a StarCraft 2 game played by what they perceived to be an AI [41], which captured explanation *demand*.

2.5 Background on RL and Our Explanation Types

We focus on model-free RL agents that learn a Q-function $Q(s, a)$ to estimate the expected future cumulative reward of taking action a in state s. After learning, the agent greedily selects actions according to Q, i.e., selecting action $\arg\max_a Q(s, a)$ in s. RL agents are typically trained with scalar rewards, leading to scalar Q-values. Thus, comparing the Q-values of two actions only provides information about which action is believed, by the agent, to achieve a larger overall future reward and how much larger. Although a human can compare the scalars to see how much the agent prefers one action over another, the scalars give no insight into the cost/benefit factors contributing to action preferences. These coarse comparisons might deny humans insight to the AI's cost/benefit tradeoffs.

2.5.1 Reward Decomposition. To help address this problem, we draw on work by Erwig et al. that exploited the fact that rewards can typically be grouped into semantically meaningful types [8]. For example, in RTS games, reward types might be "enemy damage" (positive reward) or "ally damage" (negative reward). Reward decomposition exposes reward types to an RL agent by specifying a set of types C and letting the agent observe, at each step, a $|C|$-dimensional decomposed reward vector $\vec{R}(s, a)$, which gives the reward for each type. The total scalar reward is the sum across types, i.e., $R(s, a) = \sum_{c \in C} \vec{R}_c(s, a)$. The learning objective is still to maximize the long-term scalar reward.

By leveraging the extra type information in $\vec{R}(s, a)$, the RL agent can learn a decomposed Q-function $\vec{Q}(s, a)$, where each component $\vec{Q}_c(s, a)$ is a Q-value that only accounts for rewards of type c. Using the definition of $R(s, a)$, the overall scalar Q-function can be shown to be the sum of the component Q-functions, i.e., $Q(s, a) = \sum_c \vec{Q}_c(s, a)$. Prior work has shown how to learn $\vec{Q}(s, a)$ via a decomposed SARSA algorithm [8, 44].

As described in Section 3.4.1, we use neural networks to represent the components of $\vec{Q}(s, a)$, which allows for scaling to the large state-space of our learning problem.

With SARSA, during learning, the RL executes a series of actions using an ϵ-exploration strategy and incrementally updates the Q-function components after each action. In particular, at each time step, the agent is in a particular game state s and with some small probability ϵ selects a random exploratory action and otherwise selects the greedy action. The greedy action is simply the action $\arg\max_a Q(s, a)$ that maximizes the current Q-function. After taking the selected action a in state s the agent observes the next state s' and the immediate decomposed reward vector $\vec{R}(s, a)$. This experience is used to update the decomposed Q-values $\vec{Q}(s, a)$.

The update for each component $Q_c(s,a)$ takes the form:

$$Q_c(s,a) \leftarrow (1-\alpha)\, Q_c(s,a) + \alpha\,(R_c(s,a) + \gamma Q_c(s',a'))\,,$$

where a' is the action that the RL agent will select in s'. In this update expression, $\alpha \in (0,1]$ is the learning rate, which controls how fast the Q-function adapts based on each experience, and $\gamma \in [0,1)$ is the discount factor, which specifies how much the agent should value future reward compared to immediate reward. Intuitively, the right-hand side of the update shows that $Q_c(s,a)$ is a weighted average of the current value and a target value $R_c(s,a) + \gamma Q_c(s',a')$. This has the effect of moving the current estimate in the direction of the target. The target is interpreted as an improved, but noisy, estimate of $Q_c(s,a)$ based on the most recent experience. In particular, the target is an estimates the total future discounted reward that will be achieved after executing a in s and then following the currently learned policy. This is simply the sum of the immediate reward $R_c(s,a)$ and a discounted estimate of the future total reward $Q_c(s',a')$. These SARSA-style updates are based on the classic SARSA algorithm [48] and are known to converge to the true Q-values under certain technical conditions.

Before Reference [8], others considered using reward decomposition [44, 54]—but for speeding up learning. Our focus here is on their visual explanation value. For a state s of interest, the decomposed Q-function $\vec{Q}(s,a)$ can be visualized for each action as a set of "reward bars," one bar for each component. By comparing the bars of two actions, a human can gain insight into the tradeoffs responsible for the agent's preference.

2.5.2 Saliency Visualization. Instead of the rewards, a human may want to know which parts of the agent's input were most important to the value computed for a reward (i.e., a particular $\vec{Q}_c(s,a)$). Such information is often visualized via saliency maps over the input. Our agent uses neural networks to represent the component Q-functions, letting us draw on the many neural network saliency techniques (e.g., References [46, 47, 62, 63]). While there have been a number of comparison and utility studies on such saliency maps (e.g., References [1, 3, 10, 21, 42]), there is no consensus on a best way.

Drawing from these works, we selected a saliency approach from Fong et al.'s work on image classification [9] and modified it to make it more suitable for an RTS environment. Specifically, in image applications, perturbations are applied *per pixel*, whereas our system applies perturbations *per game object*—an entire group of pixels (detailed in Section 3.4.2). Since the network may "focus" on different objects in the game for different types of reward, we compute saliency maps for each of the six decomposed reward bars, as well as the sum-total bar. These computations are made available to the UI for visualization, as we illustrate in Section 3.4.1.

3 METHODOLOGY

To investigate our research questions, we built a game and designed a between-subject four-treatment in-lab study to measure differences in how people responded to different combinations of explanations. Our 124 participants took a hands-on tutorial and then worked their way through 14 decisions the agent made, answering questions along the way. We detail each of these in the following subsections.

3.1 Game Overview

We devised our own game, where the agent controlled a kite-shaped tank, placed in the middle of four quadrants (Figure 1). We built our own game to control the action space, so people did not miss important events, which has shown up in prior research [41]. In our game, the agent had to

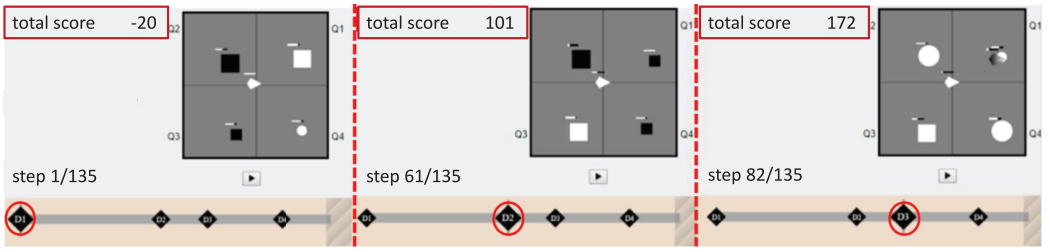


Fig. 1. A sequence of the first three DPs of the game. For each Decision Point (DP, circled in red) participants saw the game map and the score (boxed in red). Next, they made a prediction of which object the AI would choose to attack. Last, they would receive an explanation and have the ability to "play" the Decision Point. At DP1, the agent chose to attack Q2, causing a score change of 121 (+21 pts for damaging and +100 from destroying it).

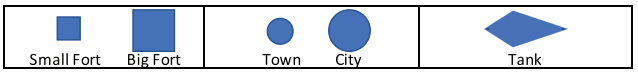


Fig. 2. The objects appearing in our game states. Enemy objects were black, and allied objects were white.

decide which of the four quadrants it would attack, and its goal was to maximize its score over each task (top-left). The game followed several rules:

- Attacking: At each DP, the agent's tank *has to* attack one of the objects. Only Forts and Tanks can attack objects (Figure 2).
- Large/small: Large forts have a higher health maximum than small forts, and large forts do more damage than small forts.
- Losing Points: If agent/friendlies (white objects) are damaged/destroyed (i.e., they lose health), then the agent loses points.
- Gaining Points: If enemies (black objects) are damaged/destroyed (i.e., enemies lose health), then the agent gains points.
- Killing: Once the agent kills something, it "respawns" on a new map, carrying over its health.
- Dying: When the agent dies, that task ends and its current score became final

By way of example, consider the transition between (Figure 1, left) → (Figure 1, middle). The agent started with −20 points, and it had to choose to attack any of the objects (either two enemies or two friends). If it wanted to attack an enemy, then it had to choose between a big fort (the big square) or a small fort (the small square). Since the agent might earn more points from the large fort, it might choose to attack it. However, if it wants to live longer to earn more points, then it might choose the small fort instead. Thus, there are tradeoffs that the agent made to try and maximize its score.

Building our own game provided the following advantages. It allowed maximum control over the entire software stack (user interface, agent API, etc.). More important, it allowed us to control the size of the action space. Popularly available RTS games have an enormous action space — Vinyals et al. estimates $\approx 10^{26}$ for StarCraft II [55]. With so many possible actions, one potential problem in investigating humans' responses to certain events or decisions in RTS games is that people observing an RTS game can become so engrossed in the game, they can miss the events or decisions of interest to the researchers (e.g., Reference [41]). By reducing the complexity of the game to a series of sequential Decision Points (DPs), we minimized the risk of people missing key events.

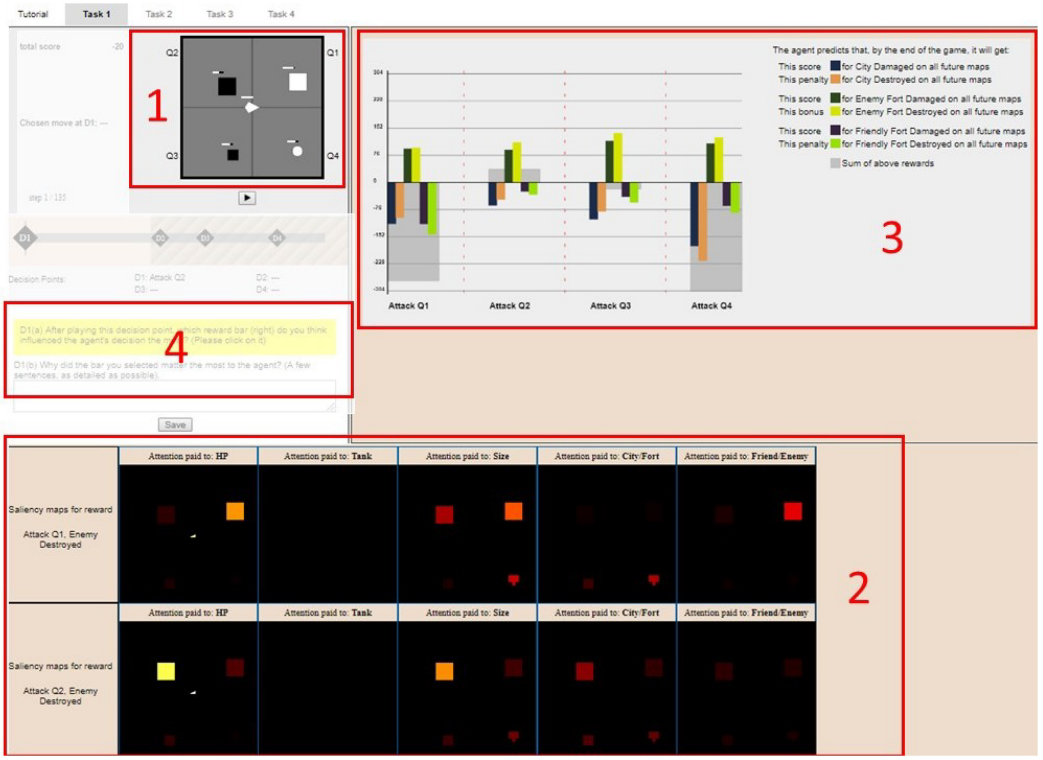


Fig. 3. A screenshot of the interface that the "Everything" participants saw. Region 1: game map, which was expanded in Figure 1. Region 2: saliency maps. Region 3: reward decomposition bars for each action. Region 4: participant question/response area.

3.2 Study Design

We performed an in-lab study using a between-subjects design with explanation style (Control, Saliency, Rewards, Everything) as the independent variable. Explanations in each explanation style (except Control, which had no explanations) were designed around Kulesza et al.'s notions of completeness and soundness in explanations [25]. Note that these notions are relative, not binary, concepts; i.e., one explanation can be more sound than another. Thus, we worked to design soundness into our explanations by avoiding approximations/simplifications of them. Similarly, we could only ensure that our explanations were more "complete" than others by making sure that every one of the agent's inputs and outputs were represented in the UI.

Our dependent variable was the quality of participants' mental models—measured by analysis of two main data sources: (1) participants' predictions of the agent's upcoming action at each Decision Point (DP), and their free-form explanations of how they made that prediction, and (2) participants' free-form answers to a post-task question about how the agent works.

We ran an ablation study, where we measured the impact of each explanation by adding or removing them, as shown in Figure 3. Thus, Everything - Rewards - Saliency = Control, as follows: (1) Control participants saw only the agent's actions, its consequences on the game state, the score, and question area (Regions 1 and 4). (2) Saliency participants saw Regions 1 and 4 plus the saliency maps (Region 2), allowing them to infer intention from the agent's "gaze" [37]. (3) Rewards participants saw Region 1 and 4 plus decomposed reward bars (Region 3), allowing them to see

Table 1. Participant Demographics per Academic Discipline

Academic Discipline	Participants	Gamers
Agricultural Sciences: 4 unique majors	8	2
Business: 3 unique majors	5	4
Engineering: 11 unique majors	63	56
Forestry: 3 unique majors	4	4
Science: 10 unique majors	25	20
Liberal Arts: 8 unique majors	9	6
Public Health and Human Sciences: 2 majors	5	1
Undisclosed	5	4
Totals	124	97

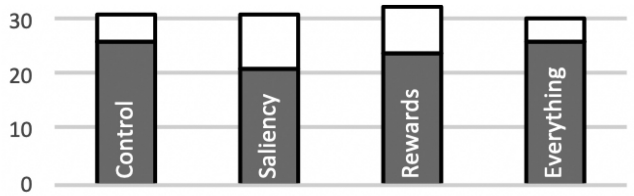


Fig. 4. Distribution of RTS "gamers" in our study. "Gamers" are shown in grey and others in white.

the agent's cost/benefit analysis for actions that it considered. (4) Everything participants saw all regions. All study materials, including the game itself, are freely available.[2]

3.3 Participants

We selected 124 participants from 208 survey respondents at Oregon State University. Since we were interested in AI non-experts, our selection criteria excluded Computer Science majors and anyone who had taken an AI course. We assigned the participants to a 2-hour in-lab session based on their availability and randomly assigned a treatment to each session.

We collected the following demographics: major and experience with RTS games (Table 1). Gamers were spread evenly across treatments such that it was unnecessary to control for this factor statistically (Figure 4).

3.3.1 Procedures. We piloted to fine-tune the experiment. Once piloting was complete, we scheduled around participant availability, using an online survey, as well as allowing for walk-in participants. Each session lasted approximately 2 hours, and each one contained between one and six participants. The treatments were randomly assigned to each session, and the participants were spread out through the lab to minimize the risk of them looking at another participants' screen. We ran a total of 55 sessions between September and October, 2018.

We began each session with a 20-minute, hands-on tutorial on the system/game, with three practice decision points. Since participants were AI non-experts, we described saliency maps as "*...like where the eyeballs of the AI fall*" and decomposed reward bars as "*...the AI's prediction for the score it will receive in the future.*" At each DP of the main tasks, participants (1) saw the game state with nothing else visible yet; (2) clicked on the object they thought the agent would attack & tell us why (Table 2); (3) upon submitting their answer, they saw what the agent did and the

[2] https://ir.library.oregonstate.edu/concern/datasets/tt44ps61c.

Table 2. The Questions We Asked Participants in All Treatments

Questions participants answered at each Decision Point (DP)
On the game map (above), which object do you think the AI will send the tank to attack? Please click on that object.
Why do you think the AI would attack this object? (A few sentences, as detailed as possible).
Asked two questions about any agent misbehaviors they saw and how they notice them.
Saliency: Asked participants which saliency map was most important and why.
Rewards: Asked participants which reward bar was most important and why.
Everything: Both of the above.
Questions participants answered after analyzing *ALL 14*DPs
Please describe the steps in the agent's approach or method. (A few sentences, as detailed as possible)
What information did you find helpful about the explanations you saw?
What information did you find problematic about the explanations you saw?
Under what circumstances is the agent likely to make a bad choice?
NASA-TLX

Questions shown in gray are not included in this analysis.

explanation for their treatment (decomposed reward bars, saliency maps, both, or neither). We recorded every object the participants clicked on and every response they gave on the screen, and the system logged these events for later analysis. Participants had 12 minutes to complete DP1 and 8 minutes per DP for the remaining 13.

After participants completed all 14 decision points, we asked them to describe the algorithm the agent used for decisions (Table 2). Collecting two metrics for mental models is consistent with Hoffman et al.'s recommendations for evaluating XAI systems [14]. After the participants completed the on-screen questions, they filled out the NASA Task-Load indeX (TLX), a validated survey to measure cognitive loads. The NASA TLX survey provided information for not only how stressful the task was but also the cognitive loads of each explanation type. Finally, participants filled out a questionnaire, which included the NASA TLX survey (described below) and the bottom question in Table 2.

3.4 Explanation Implementation

3.4.1 The Reward Decomposition Implementation. We instantiated the decomposed SARSA algorithm (Section 2.5.1) to arrive at an RL agent for our experimental domain, shown in Figure 5. The agent used six reward types to learn its decomposed Q-function $\vec{Q}(s,a)$: {Enemy Fort Damaged, Enemy Fort Destroyed, Friendly Fort Damaged, Friendly Fort Destroyed, Town/City Damaged, Town/City Destroyed}. The RL agent used a neural network representation of $\vec{Q}(s,a)$. Each component $Q_c(s,a)$ was associated with its own network, which took the game state as input and produced an estimate of $Q_c(s,a)$ as output. The input to each network represented the game state as a set of seven 40×40 greyscale images, each encoding one type of information about the game state at each of the 40×40 map locations. Specifically, five images encode the presence/absence of each of the five unit types (tanks, small forts, big forts, towns, cities), one image encodes the Health Points (HP) of the unit at each location, and one image encodes whether a unit at a location is an enemy or friendly unit. Each $Q_c(s,a)$ used the same network architecture, consisting of a single fully connected layer containing 64 hidden units with RELU activation functions.

The agent was trained for 30,000 games using decomposed SARSA (Section 2.5.1), at which point it demonstrated high-quality action choices in most situations. Training was done using a discount

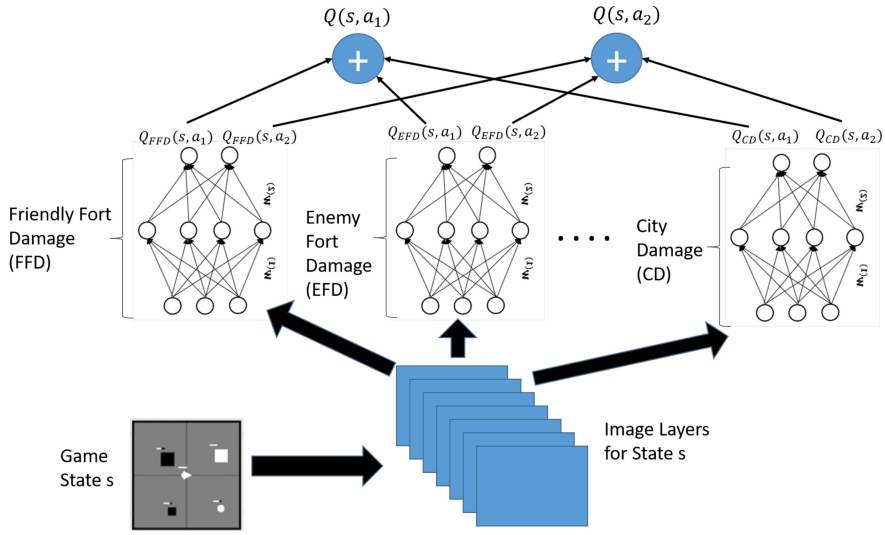


Fig. 5. RL Agent Architecture. Given a game state s, the system first transforms that state into seven layers of 40×40 images each capturing one type of game state information. The image layers are then given as input to five neural networks, one network for each reward type. Each network produces a Q-value for each action (2 actions shown for simplicity). The Q-values indicate the expected future reward of the appropriate type for taking the corresponding action. The overall Q-values for each action are the sum of the individual reward type Q-values. The RL agent selects the action with the largest overall Q-value. During training, the RL agent adjusts the parameters of the neural networks in a way that encourages the Q-values for each reward type to converge to the true values.

factor of $\gamma = 0.9$, a learning rate of $\alpha = 0.1$, and a decaying ϵ-greedy exploration strategy, where epsilon was decayed from 0.9 to 0.1.

3.4.2 The Saliency Map Implementation. Given a trained RL agent and a game state s, for each action a and each reward type c (e.g., Enemy Fort Damaged, Friendly Fort Damage, etc.), we compute a saliency map for each reward $Q_c(s, a)$. The saliency map for $Q_c(s, a)$ consists of five 40×40 images, each of which corresponds to a type of information about units in the game state: {health points, tank location, friendly/enemy, city/fort, size}. (Recall the examples in Figure 3.) Together these images indicate the spatially localized information that the agent focused on most when it computed reward $Q_c(s, a)$. As described in Section 2.5.2, we use a perturbation-based saliency approach. We implemented it as follows.

Our perturbation approach produces the saliency image for each of the five information types independently. For information type t corresponding saliency image for reward $Q_c(s, a)$ is computed as follows:

(1) Let U_t be the set of game units corresponding to information type t (e.g., for health points all game units are relevant). Starting with the current game state s produce a set of perturbed states $\{s'_u \mid u \in U_t\}$, where s'_u results from a type t perturbation of unit $u \in U_t$. For example, changing the health points of a unit. (See below for perturbation details.)
(2) For each perturbed state s'_u produce the corresponding input for the RL agent (i.e., 7 greyscale images) and pass that input to the agent to get the new value of reward $q'_u = Q_c(s_u, a)$. Note that the new RL input will only differ from the corresponding

> The agent began worried about damaging its allies…focused little on its own health and made decisions with respect to its allies…
>
> By DP3 it actually *assigned a positive point value to destroying itself in the long term because it so heavily weighted potential damage to allies*. This is because as its health dropped, it would only be able to attack allies in order to stay alive which would cause a massive penalty.
>
> Therefore, the agent decided to *always attack the largest base with the most health* so that it would take the most damage which would benefit allies in the long run. (E23)

Fig. 6. Participant E23's response to the mental model question. (This was the top scoring response to this question.) The highlighted portions illustrate both "basic" and "extra credit" concepts, described in Table 4 and Tables 9 and 10.

input for state s with respect to information type t. The saliency corresponding to unit u in s is then defined as $\Delta_u = |q'_u - Q_c(s, a)|$.

(3) Produce the saliency map for information type t by creating a map with values equal to Δ_u at locations occupied by game element $u \in U_t$. Note that, in our game, the units occupy mutually exclusive locations so that overlap does not need to be considered when producing saliency images. The saliency image for information type t has zeros at locations not occupied by units in U_t.

Thus, the intensity of a value at a saliency map location for information type t corresponds to how much the absolute value of $Q_c(s, a)$ changes when a relevant game unit is perturbed. Large output difference mean the trained RL agent was more sensitive to the perturbed part of the state—indicating importance, which we showed with a brighter color.

We chose to use a heated object scale, since Newn et al. found it to be the most understandable for their participants [36]. To make the saliency images comparable across information types, we found the maximum saliency value for each combination of reward component c and information type t in game states across 16,855 games. Normalizing each of the five images by this value ensured that each saliency image pixel's value is $\in [0, 1]$. There are many ways to map pixels to colors, but we chose to use the heated object scale (from low to high: black, red, orange, yellow, white). We chose a heat scale, because Newn et al. found it to be the most understandable way for their participants to understand gaze intensity, compared to nine other visualizations [36].

Finally, it remains to specify the perturbations used in the above procedure for each of our five information types. Each of the perturbations represented a semantically meaningful operation applied to a game unit in the game state as follows: (1) Tank Perturbation. If a tank was present, then we removed it from the game state. (2) Friend/Enemy Perturbation. Flip the friend/enemy attribute of a unit to the opposite faction. (3) Size Perturbation. Transform a unit from *big* to *small* (or vice versa). (4) City/Fort perturbations. Transform a city to be a fort (or vice versa). (5) HP Perturbation. Since HP is real valued it is treated differently. We perturbed the object HP values by a small value (30%) above and below the current HP. To produce two perturbed states. The saliency value for HP is taken to be the maximum change observed for these two states.

4 RESULTS - PEOPLE DESCRIBING THE AI

4.1 RQ-Describe Analysis Procedure

To elicit participants' understanding of the agent's decision making (RQ-Describe), we analyzed participants' answers to the end-of-session question: "*Please describe the steps in the agent's approach or method...*" [30]. (Figure 6 shows an example of one participant's response to this question.)

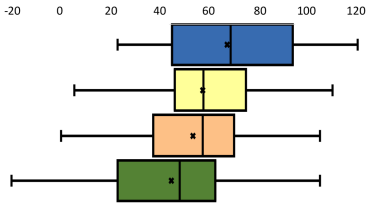


Fig. 7. The participants' final mental model scores. Box colors from top to bottom: Everything, Rewards, Saliency, and Control.

Our analysis of their responses centered on the four basic concepts the designers of the system wanted the agent to abide by the following:

(1) The agent should act consistently.
(2) The agent should act to maximize its score.
(3) The agent should attack foes, rather than friends.
(4) The agent should prefer to attack large, healthy objects to fulfill concept (2).

To devise a code set, six of the researchers (two of whom were the agent's designers) independently marked responses that they deemed potentially illustrative of participants' mental model and then discussed the results as a group. This generated an initial codeset and scoring rubric, which included the four basic concepts, for analyzing these data. Two researchers revised this initial codeset to arrive at the 18 codes shown in Tables 9 and 10. Two researchers, including one of the agent's designers), independently coded 20% of the data using these codes. That coding was, in essence, application of "ground truth"—if a participant's description was a good match to one of basic concepts the agent uses, then the matching code was assigned; otherwise, it was not. The two researchers reached 90.2% agreement [18] on the 20% they coded. Given this strong level of reliability, one researcher coded the rest.

The scoring rubric (detailed in Appendix A, Tables 9 and 10), used this code set as its basis. In the rubric each of the four main concepts listed above was assigned worth 25% of the weight—allowing responses showing understanding of only these basic concepts to reach 100% ("full credit"). Because concept (4) has two components (i.e., large and healthy), each of its components contributed half of concept (4)'s weight, i.e., 12.5% each. Subtle nuances that participants noted beyond the four main concepts earned small amounts of "extra credit" (e.g., saying the agent maximized its *future* score), whereas misconceptions not tied to the four concepts resulted in small deductions (e.g., saying it tried to preserve its HP). Weights for extra credit and deductions were categorized based on importance in 5% increments. (Codes that were merely flags had no weight; e.g., if the participant mentioned that they were uncertain.) We experimented with several different extra credit/deduction weights, and none had much effect on the resulting comparative score distributions among treatments.

Figure 7 shows the score distribution. After checking with Levene's test that the variances were not significantly[3] different ($F(3, 120) = 0.4252$, $p > 0.05$), we analyzed the data using Analysis of Variance (ANOVA).

4.2 The More, the Better?

As Figure 7 suggests, Everything participants had significantly better mental model scores than Control participants (ANOVA, $F = 8.369$, df = (1,59), $p = 0.005$) (Table 3). One possible interpretation

[3]We consider $p < .05$ as significant, and $.05 \leq p < .1$ as marginally significant by convention [6].

Table 3. ANOVA Table for Comparing the Everything Participants with the Control

	Df	Sum of Sq.	Mean Square	F-Value	Pr(>F)
Treatment	1	7847	7847.1	8.3695	0.0053
Residuals	59	55318	937.6		

This indicated that the Everything participants had a statistically significantly higher mental model score.

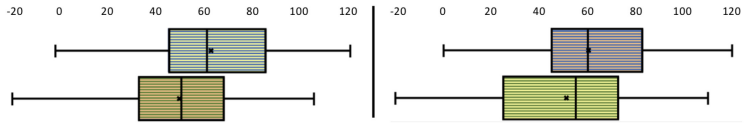


Fig. 8. Same data as Figure 7. Left: Mental model scores for those who saw *rewards* (top) and those who did not. Right: Same data, but those who saw *saliency* (top) and those who did not.

Table 4. The Four Mental Model Codes Revealing Particularly Interesting Differences in Nuances of Participants' Mental Models

Code	Count	Definition
Maximize Score	46	*The agent's overall objective is to maximize its long term score.*
Forward Looking	13	*The AI looks towards future instances when accounting for the action that it takes now.*
Paranoia	8	*The AI is paranoid about extending its life too much, expecting penalties when it should not.*
Episode Over	15	*When the AI is nearing death, it behaves differently than it has in previous decision points.*

Counts indicate the number of participants whose responses had at least one instance of a statement with that code.

is that the Everything participants' performance was due to receiving the most sound and complete explanation, consistent with Kulesza et al's results [25].

However, another possibility is that the participants in the Everything treatment were benefiting from only one of the explanation types, and that the other type was making little difference. To investigate this possibility, we isolated each explanation type as follows.

To isolate the effect of the decomposed reward bars, we compared all participants who saw them (the Rewards and Everything treatments) with those who did not. As the left side of Figure 8 illustrates, participants who saw decomposed reward bars had significantly better mental model scores than those who did not (ANOVA, $F = 6.454$, $df = (1,122)$, $p = 0.0123$).

Interestingly, isolating the effect of saliency produced a similar impact. As the right side of Figure 8 illustrates, those who saw saliency maps (the Everything + Saliency treatments) had somewhat better mental model scores (ANOVA, $F = 3.001$, $df = (1,122)$, $p = 0.0858$). This suggests that each component brought its own strengths.

4.3 Different Explanations, Different Strengths

Four of the 18 codes in our mental model codeset revealed nuanced differences among treatments in the participants' understanding of the agent. Table 4 lists these four codes.

Participants who saw rewards (Rewards and Everything) often mentioned that the agent was driven by its objective to maximize its score (Table 4's Maximize Score). Over 3/4 (36 of 46) of the people who mentioned this were in treatments that saw rewards. To consider two examples:

> (R81[4]): *The agent always tried to get as high a possible total sum of all rewards as possible. It valued allies getting damaged in the future as a rather large negative value, and dealing damage and killing enemy forts as rather high positive values.*

and

> (E14): *These costs and rewards are then summed up into an overall cost/reward value, and this value is then used to dictate the agent's action; whichever overall value is greater will be the action that the agent takes.*

Some participants who saw rewards also mentioned the nuance that the agent's interest was in its *future* score (Table 4's Forward Looking), not the present one:

> (E83): *The AI simply takes in mind the unknown of the future rounds and keeps itself in range to be destroyed 'quickly' if a future city is under attack...*

About two-thirds of the participants (9 of 13) who pointed out this nuance saw decomposed reward bars.

The agent was not perfect. One of its subtleties was its paranoia (Table 4's Paranoia). It had learned Q-value components that reflected a paranoia about receiving negative rewards for attacking its own friendly units. Specifically, even though the learned greedy policy appeared to never attack a friendly unit, unless there was no other option, the Q-components for friendly damage were highly negative even for actions that attacked enemies in many cases. Investigating this behavior revealed that it was a result of learning via the on-policy SARSA algorithm,[5] which learns while it explores.

This paranoia can be considered a type of "bug" in the agent's value estimates, so it provides an important situation for research into how to explain AI: Ideally, participants would understand the agent's behavior even (or especially!) when that behavior is flawed.

However, only eight participants showed evidence of understanding this behavior—and all 8 of them were participants who saw rewards. For example:

> (E73): *The AI appears to be afraid of what might happen if a map is generated containing four [friendly] forts or something, in which it can do a lot of damage to itself.*

However, participants who saw saliency maps (Saliency + Everything) had a different advantage over the others—noticing how the agent changed behavior when it thought it was going to die (Episode Over). For example, it tended to embark on "suicide" missions at the end of a task when its health was low. About two-thirds (10 of 15) of the participants who talked about such behaviors were those who saw saliency maps. As one participant put it:

> (S74): *If it cannot take down any structures, it will throw itself to wherever it thinks it will deal the most damage.*

4.3.1 But What Do These Nuances Cost? These results suggest that participants who saw decomposed reward bars may have had a more nuanced understanding of the agent, but their gain in understanding came with an increased cognitive load. We measured participants' perceived cognitive loads using the NASA-Task Load indeX (TLX) [12], a validated survey designed to gather

[4]First letter of participant ID is treatment (**C**ontrol, **S**aliency, **R**ewards, **E**verything).

[5]SARSA learns the value of the ϵ-greedy exploration policy, which can randomly attack friendly units. Thus, the learned Q-values reflect those random future negative rewards. However, after learning, exploration stops and friendlies are not randomly attacked.

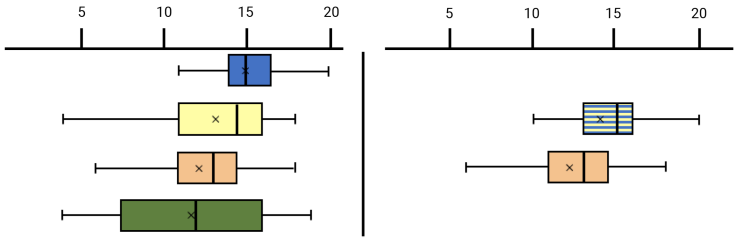


Fig. 9. Box plots of the mental demand scores. Left: The mental demand rating for Control, Saliency, Rewards, and Everything. Right: The mental demand rating for those that saw at least one explanation, i.e., those who did not see reward bars (Saliency) and those that saw the reward bars (Rewards + Everything). (Note that Control participants did not see any explanation, and thus do not appear.) The mental demand "optimal score" is zero.

participants' perceptions across six dimensions: mental demand, physical demand, temporal demand, effort, frustration, and success, using a 21-point Likert scale.

Figure 9 (left) suggested that for those participants who saw explanations, the decomposed reward bars was costly, so we visualized the comparison for those that saw decomposed reward bars in a single group against the Saliency participants (Figure 9, right). We investigated this possibility using a contrast.[6] Since we were measuring the mental demand of the *explanation*, rather than the domain, we excluded the Control participants, who did not see any explanations.

The results, shown in Figure 9 (right), showed that participants who saw decomposed reward bars (the Everything and Rewards treatments) reported a significantly higher mental demand than those who did not (the Saliency treatment) (Welch's t-test, $t(90) = 2.733$, $p = 0.007$). Further, a Hedges' effect size[7] estimate ($g = 0.54$) suggested a moderate to high practical significance. Thus, participants who saw decomposed reward bars benefited by forming a more nuanced understanding of the agent, but its cost was a significantly heavier mental demand on those participants.

4.4 Implications

On the surface, Section 4.2 suggests that, in explainable systems, the more explanation we give people, the better. However, Section 4.3 suggests that the question of which explanation or combination of explanations is better is more complex—each type has different strengths and comes at different costs, even for those with different backgrounds (Section 4.3.1 and Section C). These results suggest that there may not be a "best" explanation type across all the situations.

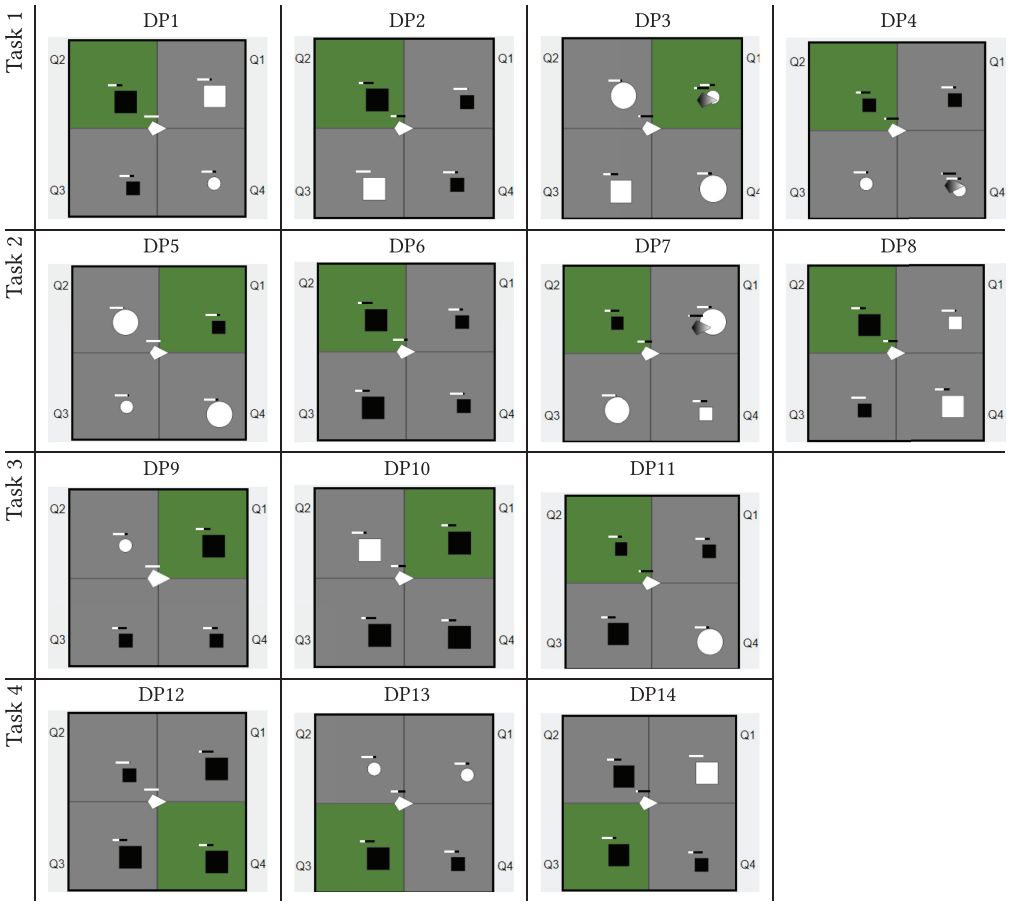
5 RESULTS: PEOPLE PREDICTING THE AI

To investigate how situational an explanation type's strength is, we turn next to how participants fared in individual situations, which we captured with their predictions at each DP.

Thus, following a similar methodology to Muramatsu et al. [34], we collected *in situ* data. Specifically, participants predicted the agent's action by clicking on the object they thought the agent would attack at every DP and then told us why they thought it would attack the object they chose (support for action predictions is discussed in Section 6) [14, 34]. Each DP's starting state is shown in Table 5, showing the quadrant the agent attacked in green. If the participants correctly

[6] A linear combination that tests: $H_0 : k_1\mu_1 + k_2\mu_2 + \cdots + k_n\mu_n = 0$, where $\Sigma_{i=1}^{n} k_i = 0$. Here, we used it to test the null hypothesis $H_0 : \frac{1}{2}(\bar{X}_R + \bar{X}_E) - \bar{X}_S = 0$ (same labeling convention as participant ID, R = Rewards, E = Everything, etc.).
[7] Used because of the unequal variances.

Table 5. The Tasks and Their Decision Points (DPs)



We have highlighted the action the AI chose in green.

predicted the action, then they received one point, receiving zero points for incorrect answers. Figure 10 shows the participants' average accuracy for each of the agent's decisions. It shows that the participants' ability to predict the agent's behavior varied widely. In this section, we present the results of a qualitative analysis that revealed several phenomena to explain why.

5.1 This Is Overwhelming

Despite having some situational advantages (Section 5.4), there were other situations where the Everything participants had the *lowest* accuracy, namely DPs 6, 9, and 11. Interestingly, the Control participants had the *highest* accuracy at these same points. This phenomenon seemed tied to Everything participants coping with two much information; not only did they have to process the reward bars (the same as the Rewards participants), they also had four rows of saliency maps they could consider (4 times as many as the Saliency participants). This highlights the importance of balancing completeness with not overwhelming users [25].

For example, some Everything participants tried to account for *all* the information they had seen. For example, at DP6:

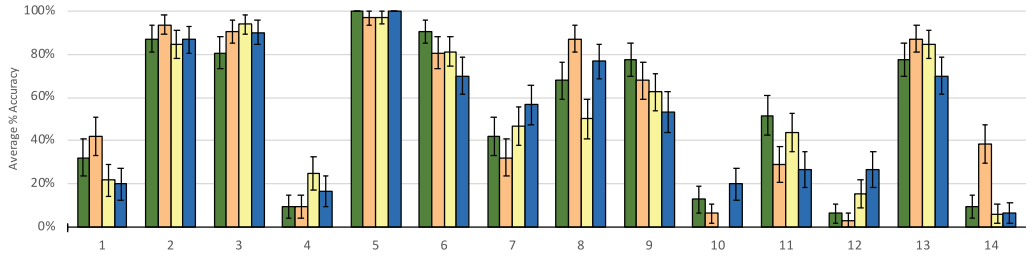


Fig. 10. Percentage of participants who successfully predicted the AI's next move at each decision point (DP), prior to receiving any explanation. Bar colors denote treatment (from left to right): Control, Saliency, Rewards, and Everything. Participants' results varied markedly for the different situations these DPs captured, and there is no evidence that any of the treatments' participants got better over time. All error bars ($SE = \sigma/\sqrt{n}$ are under 10%.

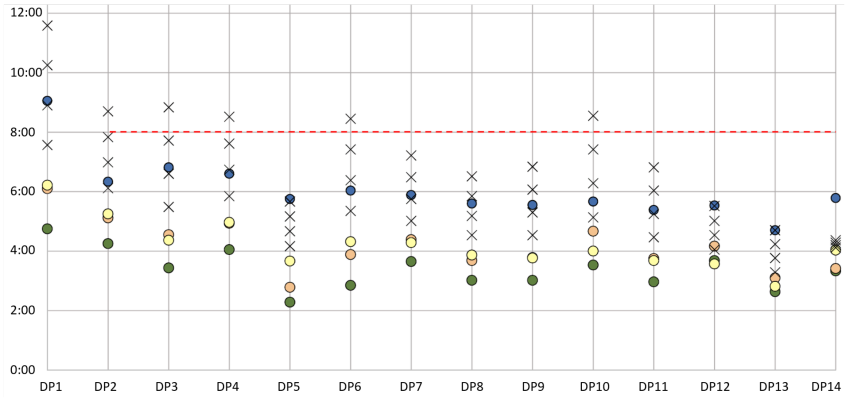


Fig. 11. Average task time vs. Decision Point (DP), per treatment. Participants had 12 minutes for the first DP and 8 minutes for all subsequent DPs. The "×"s shows the for theoretical amounts of time that the Everything participants "theoretically should have taken," depending on if they processed 1 row (the lowest ×) or 4 rows (the highest ×) of saliency maps. Notably, the lowest × is lower than the actual time they took (dark circle with highest value). Colors: same as Figure 10.

> (E38): *I think it considers own HP first then Friend/Enemy status, so going by that it will attack Q4. Also,... it attacks enemies with more HP.*

In contrast, Control participants—who saw no explanations—were able to apply simpler reasoning to form a correct prediction at DP6 (the agent attacks Q2, see Table 5):

> (C69): *because it is the lowest health of all of the enemy objects.*

Participants' timing data also attest to Everything participants' burden of processing information (Figure 11). As mentioned above, the Everything participants had to consume at most three more rows of saliency maps. Thus, to accommodate for the possibility that participants could consume as many saliency maps as they found helpful, the figure shows four "×s" for each decision point. The lowest "×" depicts how much time an Everything participant would spend if they spent as long as Control, *plus* the average time Saliency participants incurred above Control, *plus* the average time Rewards participants incurred, assuming that the Everything participants consumed as many saliency maps as the Saliency participants (i.e., at most one row). The

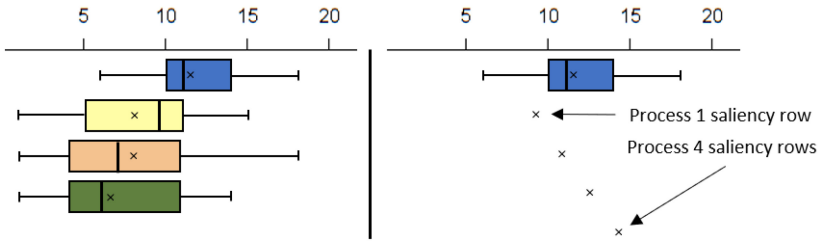


Fig. 12. Left: The participants' perceived temporal demand, where the "optimal score" is zero. The large gap between the Saliency/Rewards treatments' means and the Everything treatment's suggested that not only were Everything participants pressed for time (Figure 11), they felt it too. Right: An identical process for computing theoretical means shows the costs of considering different numbers of saliency maps.

lowest × provides a lower bound on the amount of time the Everything participants should have theoretically taken, given that they were already time constrained, if they processed the same number of saliency rows as the Saliency participants.

The *actual* time costs (Figure 11) that participants experienced were consistent with participants' *perceived* the temporal demand, which we measured using the NASA TLX (Figure 12). To compare these two measures, we performed a similar exercise as in Figure 11 to create a "theoretical" temporal perception average that the Everything participants should have felt, $\hat{\mu}_E$, by taking the temporal demand for Control (i.e., the temporal demand for the game) and adding on the demands of the decomposed reward bars and the saliency maps. Thus, the average temporal demand for the decomposed reward bars would be $\mu_R - \mu_C$, and the average temporal demand for saliency maps would be $\mu_S - \mu_C$. This means: $\hat{\mu}_E = \mu_C + (\mu_R - \mu_C) + (\mu_S - \mu_C) = \mu_S + \mu_R - \mu_C$. A paired t-test comparing $\hat{\mu}_E - \mu_E = 0$ showed a significant difference between the Everything treatment's theoretical mean and the Everything participants' actual responses (Paired t-test, $t(29) = 2.2937$, $p = 0.015$, $d = 0.59$), with Cohen's d suggesting between a moderate and large effect size.

At the end of the session, some of the participants in the Everything treatment explicitly bemoaned the complexity of processing the information, such as:

> (E39): *It was confusing all around to figure out the main factor for movement using the maps and bars…*

5.2 No Help Needed... Yet

For some DPs (2, 3, 5, 13), explanations seemed unnecessary, as the Control proved "good enough," with at least 75% of participants predicted correctly. For "Easy" situations like these, explanations may simply interfere with users' understanding; however, not everyone is likely to find the same siutations to be "easy." This suggests that explanations should be on-demand, which can provide more information to those who need it, without overwhelming those that do not.

5.3 Two Counter-intuitive Situations

In at least two kinds of situations, the agent's actions seemed counterintuitive to most participants. These situations all had very low accuracy, with *all* treatments below random guessing (≤25%).

The first situation was when the agent chose neither the strongest nor weakest of similar enemies (DPs 10,12). Some of the Everything participants got the prediction right at DP10 by combining both the saliency and the rewards explanations into their reasoning, e.g.:

> (E71): *As it will* **look at the HP of the tank more** *it will not attack Q4 instead it will go for Q1, which will* **give it enough benefit** *but also maintain its HP.*

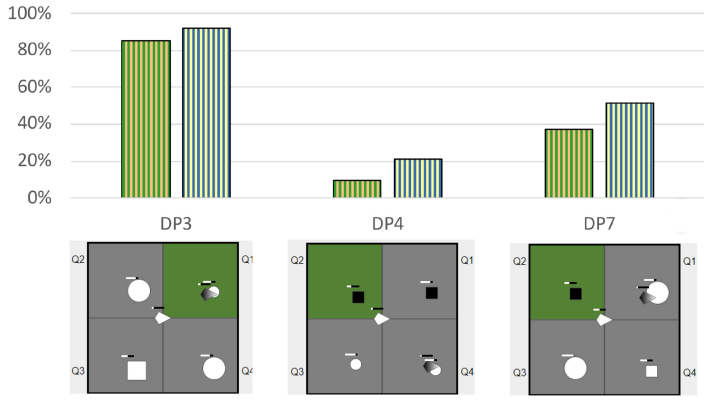


Fig. 13. A graph of all treatments' accuracy for maps that had enemy tanks. Those who did not see rewards (Control + Saliency) had lower predictive accuracy at all of these points than those who did see rewards (Rewards + Everything).

But although they were the most accurate at both of these decisions, even the Everything participants did not predict correctly better than random guessing (25%).

Worse, *all* of the participants in the Rewards treatment got DP10 wrong, suggesting that they needed the saliency maps to factor in how much the AI focuses on its *own* HP, which might have been key in this situation. For example, this Rewards participant focused on the other objects' HPs, not the agent's:

> (R94): *Lowest HP out of the 3 big fort.*

Another kind of situation that seemed counterintuitive to participants was the agent choosing to attack an enemy over saving a friend from an enemy tank (Figure 13). DP3 was the only exception to the ≤25% accuracy criteria, but the enemy tank was the *only* enemy on the map, which a majority of participants stated as the reason why the agent would attack it, such as:

> (S104): *I think it will attack this object, because it is the only enemy on the map.*

However, DP4 was the worst of such situations, with 77% of participants predicting incorrectly. Many participants incorrectly predicted it would attack the enemy tank, citing its health:

> (S18): *This is the enemy object with the lowest value for HP.*

or how threatening the enemy tank was to its friend:

> (S25): *The enemy tank poses the greatest threat [to] allies...*

Of the few participants (19 total) who predicted DP4 correctly, most (68%) were in treatments that saw decomposed reward bars, e.g.:

> (R93): *Given that the enemy tank will destroy an allied city, thus lowering points gained, the only logical decision is to remove the threat.*

5.4 A Counterintuitive Strategy Revealed: The Tanks

Despite generally low accuracy at DP4 and DP7, participants who saw decomposed reward bars seemed to have an advantageous insight into one of the agent's counterintuitive strategy choices—the agent did not care at all about enemy tanks unless the tank was the *only* enemy.

DP3, DP4, and DP7 show the three situations with enemy tanks, which the agent ignored (Figure 13: the enemy tanks are the black diamonds). The participants in the two treatments that saw decomposed rewards tended to refer to the tradeoffs that would account for these choices, using the decomposed reward bars:

> (R94): *Because destroying enemy base will give you more point than destroying a tank.*

In contrast, participants in the Control and Saliency treatments often did not realize this:

> (C69): *Because it is an enemy tank attacking a friendly city and it has low health*

and

> (S42): *before attacking you need to save your territory first.*

These three DPs were also the *only* three situations (except DP5, with 100% correctness for nearly everyone, because it was the only enemy) in which the pair of treatments with rewards explanations both produced the highest prediction scores.

This may suggest that, in an RL agent, some *strategy* choices the agent makes involving weighing potential positive/negative outcomes, may be particularly well represented by decomposed reward bar explanations.

5.5 Implications

An implication for design is that the variability in Figure 10 suggests that participants' explanation needs depended on the situation. As such, explanation systems need to cater to the user's desire for certain explanations on a case-by-case basis, allowing them not only the opportunity to gather as much (or as little) explanation as they require to update their mental model and prevent overwhelming, but also to selectively draw from the explanation they need at that moment.

An implication for research is that, from a statistical perspective, putting these different situations together would have simply "canceled each other out." (In retrospect, we view such wide variation as to be expected, given the variability in state/action pairs and the variability of human data.) However, a look at the data qualitatively produced insights that the quantitative analyses did not. This suggests the need for a mixed-methods empirical strategy to investigate XAI results that are situational. The upcoming section provides an example, in which we use a mixed-methods approach to analyze how participants formed and/or supported the predictions they made in different situations.

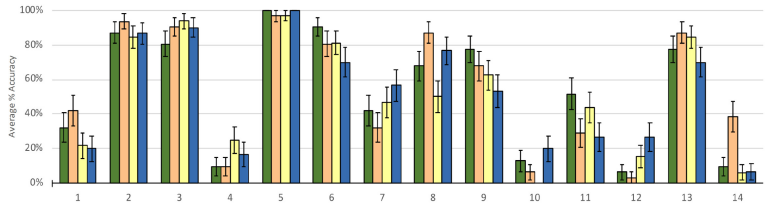
6 RESULTS: HOW PARTICIPANTS MADE THEIR PREDICTIONS

To investigate *how* participants chose the action they predicted, we used Hsieh and Shannon ([16])'s Directed Content Analysis on participants' responses to the following question: "*Why do you think that the agent will attack this object? (A few sentences, as detailed as possible).*" The interface asked this question right after each prediction participants made of what the agent would attack next, as previously shown in Table 2. Recall that the participants could not see the explanation for that agent action until *after* they answered this question. Two researchers independently coded 20% of the participants' responses using the code set in Table 6, which resulted in >80% Jaccard agreement [18]. Given this reliability, one researcher coded the rest of the corpus.

Table 6. Codeset for the Six "Prediction Why" Ways by Which Participants Explained How They Had Predicted the Agent's Next Action

Code	Definition	Example
Used Current Game State	*Participant drew on aspects of the current game state, such as object HP*	"The big fort is only about half on health..." (R80)
Used Prior Action(s)	*Participants drew on something they'd seen the agent do*	"Last time I was presented with a situation of there being less health on a tank..." (S66)
Used Prior Explanation(s)	*Participants drew on something they'd seen in the previous explanation(s)*	"In recent [saliency] maps, it seems like small size ranks higher in the Size map" (S50)
Predicted the Next Explanation	*Participants hypothesized what the explanation would look like after the agent took the action they predicted*	"I think the agent would combine the friend/enemy saliency map with the HP map...to attack the enemy fort in Q2" (S87)
Said Agent Was Random or Baseless	*Participants stated something about the agent that seemed random/wasn't made for a "good" reason*	"I don't even know anymore because this agent is rogue and insane" (S87)
Stated What *They'd* Do	*Participants placed themselves as pilot, directing what they want to attack*	"I'll attack the enemy, not my ally" (R79)

Note: *No* participant mentioned saliency maps for "Used Current Game State."



DecisionPoint	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Saliency	0	2	3	3	2	2	3	2	3	4	0	0	5	1
Rewards	8	8	7	10	4	11	11	10	9	4	8	6	6	10
Everything	6	6	3	11	6	7	7	7	9	8	8	11	6	5

Fig. 14. Top: The prediction accuracy by treatment (Figure 10). Bottom: Heatmap of number of participants who predicted the agent's *action* by predicting the system's *explanation* of that action at each Decision Point. (Vertical lines mark task boundaries.) At every decision point, more of the Rewards and/or Everything participants did this than the Saliency participants did.

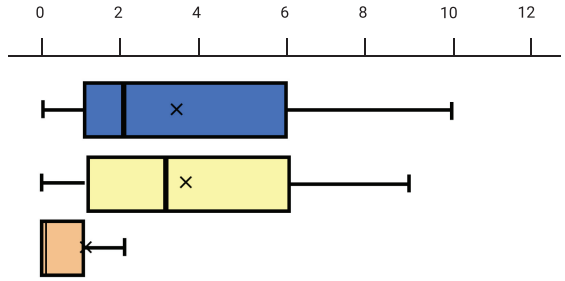


Fig. 15. Number of times participants predicted the explanations to predict the agent's actions, in the Saliency, Rewards, and Everything treatments. The variance for the Saliency participants' use of predicted explanation is tiny compared to the others.

6.1 Predicting the AI's Action by Predicting the Explanations

Interestingly, many of the participants predicted the agent's *action* by predicting how it might *explain* that action. The participants who did this were almost all participants who saw decomposed reward bars (Figure 14 and Figure 15). For example:

> (R93): *This object will provide the most damage rewards <Figure 3, Region 3, dark colored bars> and most destruction rewards <Same figure, light colored bars>*

The differences among treatments were significant: Results of Welch's ANOVA (to handle unequal variances in these data) showed a significant difference between how often the Saliency participants predicted explanations vs. those in the Rewards and Everything treatments ($F(2, 57.118) = 11.187$, $p = 0.00005$). Hedges' g test of effect sizes produced a value of 0.95, suggesting a large to very large effect size [45] of Saliency participants' versus Everything participants' rates.[8]

The Everything participants were particularly interesting, because they could see both saliency maps *and* decomposed reward bars. These participants predicted future explanations nearly as

[8]Effect size can be measured on only two populations' means. Hedges' $g = 0.99$ for Saliency participants' versus Rewards participants' rates.

Table 7. Number of Participants Who Cited Current Game State at Each DP per Treatment

Decision Point	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Control	30	26	25	24	24	28	22	25	28	27	23	26	24	27
Saliency	29	26	25	24	24	28	22	25	28	27	23	28	25	23
Rewards	25	23	20	20	23	19	18	22	19	25	18	22	18	19
Everything	27	23	27	16	21	23	21	21	21	27	20	21	21	23
Difference (%)	11.9	4.8	27.5	9.2	1.0	17.3	8.3	1.3	10.6	10.0	8.3	1.3	13.8	14.8

The highlighted number are the lowest rate of citing current game state. Notice how frequently the Rewards participants had the lowest. The "Difference" row shows the percentage difference between the lowest and second-lowest values.

often as Rewards participants did (Figure 14)—and 91% of their predicted explanations were about *only* the decomposed reward bars. Only two Everything participants predicted *both* saliency and rewards explanations. One predicted the agent's reasoning in terms of its sequence of explanations (i.e., from the "Input explanation" to the "Output explanation"):

> (E71): *It will look at if it is friend or enemy <envisioning the next saliency map> then destroy the one with the highest value <envisioning the tallest next grey reward bars> ...*

The other way was to predict the saliency explanation in combination with the rewards explanation, but without any particular order:

> (E39): *The AI will prioritize the larger target for higher bonuses <envisioning the "bonuses" small bars of the reward explanation>, focusing on size more than Hp <envisioning the size and HP portions of the saliency explanation>.*

6.2 What About Using the Game?

When participants were looking in one direction, they often were not looking in another (Table 7). The highlighted portions show the treatment that had the lowest rate of using the current game state information. Whenever the Everything participants were the lowest, there was only an average difference of 3.2% between them and the second-lowest treatment's participants (which were the Rewards at all 4 instances). However, the other 10 decisions, the rate that the Rewards participants mentioned current game state *was* the lowest, and the average difference between their rate and the second-lowest jumped to 10.6%.

In fact, the average rates of using current game state for the four treatments were very different (Figure 16). Levene's test revealed that the variances were not significantly different ($F(3,120) = 2.295$, $p > 0.05$). ANOVA showed that there was at least one mean significantly different than the others ($F(3,120) = 6.0481$, $p = 0.0007$) (Table 8). Removing the Rewards treatment got rid of the significance (ANOVA, $F(2,89) = 2.1987$, $p = 0.117$), which meant that it was the Rewards treatment's participants that had a statistically significantly lower reliance on current game state than all of the other treatments.[9]

6.3 Predict-How: Discussion

These results showed that participants had two main differences in how they supported their predictions: Their use of the current game state without mentioning saliency maps and forming

[9]The averages for all six codes by treatment are in Table 11.

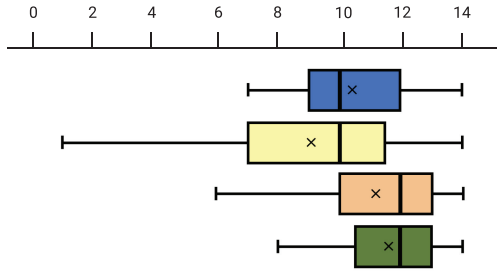


Fig. 16. Box plots for the number of times participants used the current game state in the Control, Saliency, Rewards, and Everything treatments. Note that participants who saw rewards had a lower frequency of citing current game state information in their justification.

Table 8. ANOVA Table for Current Game State across All Treatments

	Df	Sum of Sq.	Mean Square	F-Value	Pr(>F)
Treatment	3	113.05	37.683	6.0481	0.0007
Residuals	120	747.66	6.231		

This indicated that at least one of the means was different for the four treatments.

predicted explanations. Reward participants demonstrated a tradeoff between using predicted explanations (which they did significantly more than the Saliency participants) and using current game state (which they did significantly less than all the others). Figure 15 also suggested that the Everything participants were engaging in this exercise, but we discovered that 91% of their responses pertained to *only* decomposed reward bars. However, there are a couple of caveats to this.

Even though the Rewards and Everything participants predicted explanations at a higher rate than the Saliency participants, it did not necessarily help them to predict the *agent's action*. This is why these two treatments had the highest accuracy at *only five* total DPs—three of which had a common theme (Section 5.4). However, the Saliency participants were also the most accurate at five of the DPs, but they had an entirely different approach, using current game state information almost as frequently as the Control participants (Table 7). Thus, even though the decomposed reward bars allowed participants a more nuanced understanding of the agent's reasoning (Section 4), and the participants gained strategic insights from them (Section 5.4), it remains unclear whether they definitively helped participants predict the agent's actions.

Well-known HCI textbooks, like Reference [19], recommend placing information in the UI close together to make sure people notice them. However, Figure 3's Regions 1 and 3, which show game state information and the decomposed reward bars per action, respectively, are difficult to position such that they are both proximate and understandable. Although we did not gather gaze-tracking data, it could have been the case that the Rewards participants spent "too much time" looking in Region 3, since they only had a limited amount of time per DP. This attention expenditure was probably worse for the Everything participants, since they had to choose whether to consume the decomposed reward bars and/or the saliency. However, their habit of hypothesizing *just* the decomposed reward bars implies that they might have spent most of their time in Region 3, too. In designing this system, we had to trade off between being complete [25] and placing information proximate [19]. Note that our domain's action space was only 4, as opposed to Vinyals et al.'s approximation for StarCraft II's action space of 10^{26} [55]. A challenge for explainable systems,

especially as the action space grows larger (and not just in the RTS domain), will become how to effectively manage information in such a way to adhere to good design principles.

7 THREATS TO VALIDITY

Every study has threats to validity [59]. For our study and analysis, we addressed the impact of participants' experience with RTS games and engineering background on their mental models, but we did not collect additional demographics that would have allowed considering other factors that may impact people's mental models of games, such as age. Our study design also emphasized isolation of variables over external validity. For example, we simplified the game to avoid confounding factors, such as participants missing events or using different points of view [41], but this means that our results on how effective the explanations are for this small action space might not hold for complex RTS games; thus, we do not know how effectively these explanations will scale when the action space is much larger. We also controlled participants' time per DP, but this may have impacted their mental models with insufficient time per DP (8 minutes) or too few DPs (14).

The Everything treatment saw more granular saliency explanations than the Saliency treatment did. Specifically, Everything participants could see a saliency map row for *each* reward type in the reward bars via the 2D *grid* of saliency (Figure 3, Region 2), whereas participants in the Saliency treatment saw only one row of saliency, which mapped to their one-item view of rewards (the winning action). That is, Saliency participants could observe one "output"—the action—so they saw a row of "inputs" for that output, whereas Everything participants saw four potential "outputs," so they saw 4 rows of "inputs." This design decision helped us to ensure that the mappings from inputs to outputs made sense, but introduced an uncontrolled difference in the saliency map presentation the Saliency treatment saw from the saliency map presentation the Everything treatment saw. We attempted to address this in Section 5.1, Figures 11 and 12 by introducing a constant scaling factor for the number of saliency rows processed, but this only provides a lower-bounded estimate that could only be measured by gathering gaze-tracking data of every participant, which we did not do.

In this game, with only four possible actions, there were 24 decomposed reward bars on the screen, and each one had five saliency maps. Thus, it is not clear whether our results would scale up to more complex games like StarCraft II, since it involves hundreds of potential decisions to make at any given moment.

Some participants used phrasing that rendered their reasoning unclear to us. For example, when participants referred to an object as "biggest threat," we could not determine whether they were referring to something that they observed on the game map (e.g., the unit with the largest HP/attack damage as the "biggest threat") or just intuition (e.g., a gut feeling that the "biggest threat" should have a scary appearance). Another example of unclear reasoning was when they based their decisions on the distance between DPs on the timeline (e.g., if a DP was short, choosing the object with the smallest health and giving the timing as evidence for their choice).

The Control participants—who could rely only on previous state or past actions—demonstrated a remarkable similarity to the Saliency participants in the information they cite to support their choices (a 3% difference). However, this might have been due to the fact that the Saliency participants might have been using saliency map information in such a way that we could not detect. For example,

(S61): *It's a small enemy fort with relatively lower HP*

was coded with *Current Game State* information, but by using size and HP to support their choice (both represented by saliency maps), this Saliency participant might have been using predicted saliency map information to identify where the agent might focus on parts of the game state when

forming their answer—without us being able to detect it. A way to mitigate this in future work would be to provide ways for participants to select the information they want to use and then describe how that information looks.

Threats like these can only be addressed with more studies using diverse empirical methods to generalize the findings.

8 CONCLUDING REMARKS

In this article, we report on a mixed methods study with 124 participants with no AI background. We investigated which of four visual explanation possibilities—saliency maps, decomposed reward bars, both, or neither—enabled participants to build the most accurate mental models, and in what circumstances.

Our results do not point to any one *best* explanation for helping people form mental models, but rather that:

- Everything participants provided significantly better mental model description than the Control participants (Section 4).
- Rewards participants offered the most insight into nuanced concepts, such as agent paranoia (Section 4).
- Each of the Saliency, Rewards, Everything, and even the Control treatments outperformed the others multiple times (Section 5).
- Although counterintuitive behaviors by the agent were difficult for *everyone* to understand, participants who saw rewards (i.e., Rewards and Everything participants) showed some advantages in understanding these counterintuitive behaviors (Section 5).
- Some participants, especially those in the Rewards treatment, predicted the agent's next *action* by predicting its next *explanation* (Section 6).
- Participants in the Rewards and Everything treatments sometimes experienced high cognitive loads and reported feeling overwhelmed. This was especially common in the Everything treatment (Section 5).

Abstracting above these results may add to the design takeaways we can derive from them. Toward this end, we turn again to Lim and Dey's work on intelligibility types [27] as interpreted and extended by Dodge et al. [7]. These works provide a way of semantically categorizing different types of explanations in terms of the type of content provided.

We have already pointed out that the decomposed reward explanations are a form of the "Outputs" intelligibility type (which explain how outputs are being selected) [27], and the saliency map explanations are a form of the"Inputs" intelligibility type (which explain which inputs are being considered) [27]. Our results also help to point out what the perspective a third type can bring: the "What" intelligibility type (for RTS games, anything about game state [7]).

What the "What" explanation results suggest is that *anything* in the environment that helps to explain how the AI is reasoning is itself an explanation. The results in Section 6 help to demonstrate this point. In that section, in which participants described how they were figuring out the AI by answering "Why do you think that the agent will attack this object," at least half the participants in every treatment integrated state information into their responses. This suggests more generally that people may mentally integrate "What" information with other types of explanations that are provided by the system—i.e., that text, graphics, and animations in the state and environment are functioning as *part* of the explanation. This in turn suggests that XAI researchers may be

well served by treating *all* information in the environment as part of the explanation, in both the explanation design and explanation evaluation components of their research.

Our results also corroborate Lim et al.'s position that each intelligibility type has different strengths, based on the explanation goal of the user [29]. The user's goals for an explanation also depend on the situation, and Section 5's result that every explanation outperformed the others in some situations helps to demonstrate this. Likewise, each intelligibility type has different weaknesses, and Section 5's result that every explanation sometimes *under*performed compared to the others helps to demonstrate this point.

Indeed, our analyses suggest several one-size-does-*not*-fit-all takeaway messages. First, one type of explanation does not fit all *situations*, as Section 5 shows. Second, one type of explanation does not fit all *people*, as the distribution ranges in Figure 7 show. And perhaps most critical, one type of empirical analysis (strictly quantitative or strictly qualitative) was not enough; only by combining these techniques were we able to make sense of the wide differences among individual participants at individual DPs. We believe that, only by our community applying an arsenal of empirical techniques, can we gain the rich insights needed to learn how to explain AI effectively to mere mortals.

APPENDIXES

A MORE ON MENTAL MODEL QUALITATIVE CODE SET

As described in Section 4, we generated a codeset for participant responses to the summary question, "*Please describe the steps in the agent's approach or method...*" [30]. To do so, we created an initial codeset at a large group meeting with most of the stakeholders. Two researchers then worked toward agreement, revising the codeset along the way. Upon reaching agreement—but before coding the full data corpus, the same large group meeting established rubric point values for the presence or absence of each code. The final codeset is shown in Tables 9 (all positive valued codes) and 10 (all negative valued codes. Along with the codes and the definitions, we provide the rubric value for each code (in parentheses with the code name).

B MORE ON PREDICTION WHY QUALITATIVE CODE SET

In Section 6, we described the differences in the means for two of the codes. Table 11 shows the averages for all of the codes. In our data, those two codes were the only interesting differences that we noted.

C PARTICIPANT BACKGROUND: DID IT MATTER?

Of the participants, 51% were in an engineering major, and 77% of them had more 10+ hours of RTS game experience, so we investigated whether these backgrounds helped them even if they had no AI/ML background, such as contributing to their abilities to reverse-engineer complex systems or their knowledge of "good" strategy.

Table 12 suggests that for Description scores, the participants in the Control and Saliency treatments benefited demonstrably from their backgrounds in gaming and engineering. However, for the other two treatments (Rewards and Everything) the correlation ranged from small to no *negative* effect. For their Prediction scores (Table 13), the correlations for predicting the agent ranged from no effects to small positive and negative effects in the four treatments.

Table 9. Positive Codeset Used for Evaluating Mental Model Quality

Code	Definition	Example from participant responses
Consistency (25)	AI is an algorithm, and enacts some consistent policy. May include attempts to generalize rules from their observations OR explicitly mentions that the policy may be too complex to generalize	The AI always picked an Enemy object when given the choice between Friend and Enemy.
Score Max. Objective (25)	AI maximizes score or is greedy. Assigning a score and basing a decision upon it is considered an implicit maximization	I believe that the AI looks at scenarios that would make it get the highest amount of points.
Foe > Friend (25)	AI prefers to attack enemies and not friends or that identifying friends and foes. Kind of treat this as a keyword search for [friend, enem(ies,y)]. However, this MUST be posed as a general rule. There may be some implicit ways they make it clear they do understand friendly fire	The AI basically attacked any enemy that has the highest total score in all the three steps as shown above.
Large > Small (12.5)	The AI generally prefers large forts to small ones. They need to either use size words, or map whatever else they use (threat, DPS) onto size concepts	otherwise it attacks big forts then little forts giving preference to the lower health of each species.
High HP > Low HP (12.5)	The AI generally prefers attacking enemies with higher HP. For all these, we need to see it posed as a general rule NOT that a single decision was made this way	Therefore, the agent decided to always attack the largest base with the most health so that it would take the most damage that would benefit its allies in the long run.
Distance Concept Used (0)	There are circumstances where distance matters, but not usually. If they mention it as being highly, rarely, or never relevant, code it.	When it comes to the agent's approach, a general rule that it likes to follow is to attack the closest enemy.
Episode Over (5)	AI detected when it cannot survive and is reckless at last step	If it cannot take down any structures, then it will throw itself to wherever it thinks it will deal the most damage
Specific decision citation (0)	Reference to specific decision point to develop mental model. Could be by task/decision index, or by some description of the game board	In Task 1 D3, the only enemy is a tank, so the tank is attacked by the AI

(Continued)

Table 9. Continued

Code	Definition	Example from participant responses
Fort > Tank (10)	AI does not preserve allies very much OR declines to attack the enemy tank	it doesnt attack enemy tank sometimes when its clearly preoccupied and at lowest health(maybe enemy tank is an exception or this is an incorrect prediction)?
Forward Looking (5)	Participant makes it explicit that the agent is taking actions to make future actions easier (or available)	Specifically, it seems that the agents is simulating not just the immediate outcome of an action, but also the outcomes down the road, given all the possible other choices it may have next
Damage Racing (5)	AI does not preserve allies, because it may not have to pay the penalty if it can end the simulation before it dies	The AI simply takes in mind the unknown of the future rounds and keeps itself in range to be destroyed ”quickly” if a future city is under attack even at the expense of having more HP (and opportunity for damage and thus points) in future rounds.
Specific game actions have the following specific point values	the attack damage for the smaller objects with a ”low” damage do half the damage as the tank (which also is labeled as a ”low” damage object) and the objects with a ”high” damage output	
Paranoia (10)	Identified that the agent is paranoid about extending its life too long, since it thinks it will encounter maps with all friendlies OR a map where an enemy tank will kill something.	Since Q4 has more health than Q1 or Q3, it is worth more total points, and the resulting low health of the AI tank translates into fewer potential scenarios where the tank damages or destroys friendly targets. The AI does not know what the next map will look like, it only knows what health the tank will have carrying forward, and I think it only calculates out point potentials for this particular tank.

The rubric used the parenthesized weights (first column) to add to participants' mental model scores.

Table 10. Negative Codeset Used for Evaluating Mental Model Quality

Code	Definition	Example from participant responses
Answered only specifically (−2)	Participant did not attempt to generalize a rule, only mentions some specific decisions	The AI decided to choose between Q4 and Q3 although it should have been Q2 and Q4. From Q4 and Q3, it picked Q3, which make me wonder why, since it gives way less points.
Uncertainty (0)	articipant is unsure about something with respect to how the agent works OR poses a question about the agent (possibly speculatively)	As to how the value of the rewards is determined by the AI I don't know and I think may be vulnerable to flaws.
Score "feature" (−5)	Putting "score" in a priority list with things like Type, which are not commensurate (it is not an input to the system, but the quantity we are maximizing over)	It generally seemed to think that scoring points was more important than preserving the life of its allies, or even the life of itself. ... It made decisions based on score, and it oftentimes felt like it made decisions based on score alone.
Passive fort threat (−5)	Participant perceives forts that will do literally nothing as threatening to the agent or its allies. To clarify, when the agent attacks Q2, enemy forts in other quadrants won't do anything.	It had to attack, so it based its risk assessment on the fact that sometimes there was a big threat it had to overcome to preserve its allies, or sometimes it saw that those allies could survive and so went for something else that it could attack.
Rules Misconception (−5)	Misunderstood something about what is possible within the rules of the game. Might be one of the above "named" misconception, or somethig miscellaneous	Goal #2 is checking to see if it can defeat multiple objects by focusing certain objects
Incorrect Objective (−15)	As opposed to maximizing score, the participant expresses belief that it is doing something else. Examples include minimizing threat, behaving arbitrarily, or attacking the nearest object.	The AI, for the most part, seemed to prioritize destruction of the closest enemy object rather than strategic destruction of enemies or preservation of ally objects.
Cherry Picking (−5)	AI focuses on things with low hp (or small type), which can be easily defeated	The decision to attack Q4 first comes from the current HP of the small fort. It is low enough that it results in a quick elimination.

(Continued)

Table 10. Continued

Code	Definition	Example from participant responses
self Preservation (−15)	AI preserves the tank as much as possible (it does not)	if no allies are present, it will attack the enemy that will preserve the AI's tank the most, while also trying to destroy enemies
Ally Preservation (−5)	AI preserves its allies	It also worked sometimes for self-preservation, but as it went on, it seemed more concerned with protection versus self-preservation.
Assigning value based on social biases (0)	Thinking that the AI should defend its friends or do things a human would, often leading to a negative value judgment about the agent	This is the sort of thinking I have seen in players who only take risks when the deck is stacked in their favor.
Agent Behavior Misconception (−5)	Something not quite right about their understanding of what the agent is doing or why it is doing it. Similar to rules misconception, could be "named" or miscellaneous	The AI wants to do damage control and will almost always pick the attack where this penalty score is the lowest.... In this case, it would pick the largest positive score.
Explanation Misconception (−5)	Misunderstood something SPECIFIC about the explanation contents	So the AI tries to construct reward bars by predicting attributes of the objects in each quarter.
Agent Behavior Misconception (−5)	Something not quite right about their understanding of what the agent is doing or why it is doing it. Similar to rules misconception, could be "named" or miscellaneous	The AI wants to do damage control and will almost always pick the attack where this penalty score is the lowest.... In this case, it would pick the largest positive score.
Explanation Misconception (−5)	Misunderstood something SPECIFIC about the explanation contents	So the AI tries to construct reward bars by predicting attributes of the objects in each quarter.

Weights for each code are given in parentheses in the first column. The rubric used the parenthesized weights (first column) to subtract from participants' mental model scores. Note that some of these codes have 0 rubric value, as we were just using them as flags.

Table 11. Average Number of Times That Participants Used Each Code

Treatment Name	Cited Current Game State	Used Prior Action(s)	Used Prior Explanation(s)	Formed Predicted Explanation	Cited Random Choice	Stated What They'd Do
Control	11.6	1.7	—	—	0.03	0.1
Saliency	11.2	1.8	1.4	1.0	0.06	0.3
Rewards	9.1	1.6	1.1	3.5	0.0	0.4
Everything	10.4	1.2	1.0	3.3	0.0	0.0

Notice the differences for Cited Current Game State and Formed Predicted Explanation.

Table 12. Pearson's Correlation, r, between the Mental Model Description Scores and Having 10+ Hours of Gaming Experience (Column 1) and Being in the School of Engineering (Column 3)

Treatment	% Participants Gamers	% Participants Engineering	Correlation: Gamer & Description Score	Correlation: Engineering & Description Score
Control	84%	39%	0.44	0.25
Saliency	68%	39%	0.31	0.10
Rewards	75%	59%	−0.02	−0.14
Everything	87%	66%	−0.19	−0.10
Overall	78%	51%	0.15	0.08

Across all participants, there was a small effect from being a gamer, but for those that did not see decomposed reward bars (Control + Saliency), their gaming background helped them more. Similarly, for the school of engineering, there was overall no effect on their ability to describe the algorithm.

Table 13. Pearson's Correlation, r, between the Participants' Scores and Having 10+ Hours of Gaming Experience (Column 1) and Being in the School of Engineering (Column 2)

Treatment	% Participants Gamers	% Participants Engineering	Correlation: Gamer & Prediction Score	Correlation: Engineering & Prediction Score
Control	84%	39%	0.09	−0.18
Saliency	68%	39%	0.11	0.01
Rewards	75%	59%	0.14	−0.14
Everything	87%	66%	−0.06	−0.24
Overall	78%	51%	0.05	−0.17

Overall, there was no correlation for gaming experience and a small negative correlation for being in the school of engineering.

REFERENCES

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, Vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 9505–9515.

[2] Dan Amir and Ofra Amir. 2018. HIGHLIGHTS: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1168–1176.

[3] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *Proceedings of the International Conference on Learning Representations*.

[4] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: A visual interactive hybrid recommender system. In *Proceedings of the ACM Conference on Recommender Systems*. ACM, 35–42.

[5] Michelene T. H. Chi, Miriam Bassok, Matthew W. Lewis, Peter Reimann, and Robert Glaser. 1989. Self-explanations: How students study and use examples in learning to solve problems. *Cogn. Sci.* 13, 2 (4 1989), 145–182. DOI: https://doi.org/10.1207/s15516709cog1302_1

[6] Duncan Cramer and Dennis Howitt. 2004. *The Sage Dictionary of Statistics: A Practical Resource for Students in the Social Sciences*. Sage.

[7] Jonathan Dodge, Sean Penney, Claudia Hilderbrand, Andrew Anderson, and Margaret Burnett. 2018. How the experts do it: Assessing and explaining agent behaviors in real-time strategy games. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'18)*. ACM, New York, NY, Article 562, 12 pages.

[8] Martin Erwig, Alan Fern, Magesh Murali, and Anurag Koul. 2018. Explaining deep adaptive programs via reward decomposition. In *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence*. 40–44.

[9] Ruth Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*. IEEE, 3449–3457.

[10] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. 2018. Visualizing and understanding Atari agents. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholm Sweden, 1792–1801.

[11] Sandra G. Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. Sage, Thousand Oaks, CA, 904–908.

[12] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In *Advances in Psychology*. Vol. 52. Elsevier, 139–183.

[13] Bradley Hayes and Julie A. Shah. 2017. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 303–312.

[14] Robert Hoffman, Shane Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable AI: Challenges and prospects. *arXiv:1812.04608* (2018).

[15] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. 2019. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 579.

[16] Hsiu-Fang Hsieh and Sarah Shannon. 2005. Three approaches to qualitative content analysis. *Qual. Health Res.* 15, 9 (2005), 1277–1288.

[17] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. 2019. Enabling robots to communicate their objectives. *Auton. Robots* 43, 2 (2019), 309–326.

[18] Paul Jaccard. 1908. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.* 44 (1908), 223–270.

[19] Jeff Johnson. 2013. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Elsevier.

[20] Caitlin Kelleher and Wint Hnin. 2019. Predicting cognitive load in future code puzzles. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'19)*. ACM, New York, NY.

[21] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholm Sweden, 2668–2677. http://proceedings.mlr.press/v80/kim18d.html

[22] Man-Je Kim, Kyung-Joong Kim, SeungJun Kim, and Anind K. Dey. 2016. Evaluation of starcraft artificial intelligence competition bots by experienced human players. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1915–1921.

[23] M. J. Kim, K. J. Kim, S. Kim, and A. K. Dey. 2018. Performance evaluation gaps in a real-time strategy game between human and artificial intelligence players. *IEEE Access* 6 (2018), 13575–13586. DOI: https://doi.org/10.1109/ACCESS.2018.2800016

[24] Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educ. Psychol.* 41, 2 (2006), 75–86.

[25] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI'15)*. ACM, 126–137.

[26] 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 *Proceedings of the 2013 IEEE Symposium on Visual Languages and Human Centric Computing*. IEEE, 3–10.

[27] Brian Y. Lim. 2012. *Improving Understanding and Trust with Intelligibility in Context-aware Applications*. Ph.D. Dissertation.

[28] Brian Y. Lim and Anind K. Dey. 2009. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th International Conference on Ubiquitous Computing*. ACM, 195–204.

[29] Brian Y. Lim, Qian Yang, Ashraf M Abdul, and Danding Wang. 2019. Why these explanations? Selecting intelligibility types for explanation goals. In *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI) Workshops*.

[30] Katherine Lippa, Helen Klein, and Valerie Shalin. 2008. Everyday expertise: Cognitive demands in diabetes self-management. *Hum. Fact.* 50, 1 (2008), 112–120.

[31] M. Lomas, R. Chevalier, E. V. Cross, R. C. Garrett, J. Hoare, and M. Kopack. 2012. Explaining robot actions. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI'12)*. 187–188. DOI: https://doi.org/10.1145/2157689.2157748

[32] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. 2019. Explainable reinforcement learning through a causal lens. *CoRR* abs/1905.10958 (2019). arxiv:1905.10958 http://arxiv.org/abs/1905.10958

[33] Tim Miller. 2017. Explanation in artificial intelligence: Insights from the social sciences. *CoRR* abs/1706.07269 (2017). arxiv:1706.07269 http://arxiv.org/abs/1706.07269

[34] Jack Muramatsu and Wanda Pratt. 2001. Transparent queries: Investigation users' mental models of search engines. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 217–224.

[35] NASA. [n.d.]. NASA TLX: Task Load Index. Retrieved from https://humansystems.arc.nasa.gov/groups/TLX/.

[36] Joshua Newn, Eduardo Velloso, Fraser Allison, Yomna Abdelrahman, and Frank Vetere. 2017. Evaluating real-time gaze representations to infer intentions in competitive turn-based strategy games. In *Proceedings of the ACM Symposium on Computer–Human Interaction in Play*. ACM, 541–552.

[37] Joshua Newn, Eduardo Velloso, Marcus Carter, and Frank Vetere. 2016. Exploring the effects of gaze awareness on multiplayer gameplay. In *Proceedings of the ACM Symposium on Computer–Human Interaction in Play Companion Extended Abstracts*. ACM, 239–244.

[38] Donald Norman and Dedra Gentner. 1983. Mental models. *Lawrence Erlbaum Associates, Hillsdale, NJ*,1983.

[39] S. Ontañón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss. 2013. A survey of real-time strategy game AI research and competition in StarCraft. *IEEE Trans. Comput. Intell. AI Games* 5, 4 (Dec. 2013), 293–311. DOI: https://doi.org/10.1109/TCIAIG.2013.2286295

[40] Bei Peng, James MacGlashan, Robert Loftin, Michael L. Littman, David L. Roberts, and Matthew E. Taylor. 2016. A need for speed: Adapting agent action speed to improve task learning from non-expert humans. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 957–965.

[41] Sean Penney, Jonathan Dodge, Claudia Hilderbrand, Andrew Anderson, Logan Simpson, and Margaret Burnett. 2018. Toward foraging for understanding of StarCraft agents: An empirical study. In *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI'18)*. ACM, NY, 225–237.

[42] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. 2013. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proceedings of the IEEE International Conference on Computer Vision*. 1153–1160.

[43] Ariel Rosenfeld, Moshe Cohen, Matthew E. Taylor, and Sarit Kraus. 2018. Leveraging human knowledge in tabular reinforcement learning: A study of human subjects. *Knowl. Eng. Rev.* 33 (2018).

[44] Stuart Russell and Andrew Zimdars. 2003. Q-decomposition for reinforcement learning agents. In *Proceedings of the International Conference on Machine Learning*. 656–663.

[45] Shlomo S. Sawilowsky. 2009. New effect size rules of thumb. *J. Mod. Appl. Stat. Methods* 8, 2 (2009), 26.

[46] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR* abs/1312.6034 (2013).

[47] Jost Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2014. Striving for simplicity: The all convolutional net. *CoRR* abs/1412.6806 (2014).

[48] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

[49] Rohani A. Tarmizi and John Sweller. 1988. Guidance during mathematical problem solving.*J. Educ. Psychol.* 80, 4 (1988), 424.

[50] Andrea L. Thomaz, Guy Hoffman, and Cynthia Breazeal. 2006. Reinforcement learning with human teachers: Understanding how people want to teach robots. In *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN'06)*. IEEE, 352–357.

[51] J. Tullio, A. Dey, J. Chalecki, and J. Fogarty. 2007. How it works: A field study of non-technical users interacting with an intelligent system. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 31–40.

[52] Jasper van der Waa, Jurriaan van Diggelen, Karel van den Bosch, and Mark Neerincx. 2018. Contrastive explanations for reinforcement learning in terms of expected consequences. (2018).

[53] Jeroen J. G. Van Merrienboer and John Sweller. 2005. Cognitive load theory and complex learning: Recent developments and future directions. *Educ. Psychol. Rev.* 17, 2 (2005), 147–177.

[54] Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. 2017. Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*. 5392–5402.

[55] Oriol Vinyals, David Silver, et al. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. Retrieved from shorturl.at/dinL3.

[56] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y. Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'19)*, Vol. 19.

[57] Daniel S. Weld and Gagan Bansal. 2019. The challenge of crafting intelligible intelligence. *Commun. ACM* 62, 6 (May 2019), 70–79. DOI: https://doi.org/10.1145/3282486

[58] Ari Widyanti, Addie Johnson, and Dick de Waard. 2013. Adaptation of the rating scale mental effort (RSME) for use in Indonesia. *Int. J. Industr. Ergon.* 43, 1 (2013), 70–76.

[59] Claes Wohlin, Per Runeson, Martin Höst, Magnus Ohlsson, Björn Regnell, and Anders Wesslén. 2000. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Norwell, MA.

[60] Robert H. Wortham, Andreas Theodorou, and Joanna J Bryson. 2017. Improving robot transparency: Real-time visualisation of robot AI substantially improves understanding in naive observers. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (ROMAN'7)*. http://opus.bath.ac.uk/55793/

[61] Johannes Zagermann, Ulrike Pfeil, and Harald Reiterer. 2016. Measuring cognitive load using eye tracking technology in visual computing. In *Proceedings of the 6th Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*. ACM, 78–85.

[62] Matthew Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*. Springer, 818–833.

[63] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2018. Top-down neural attention by excitation backprop. *Int. J. Comput. Vis.* 126, 10 (Oct. 2018), 1084–1102. DOI: https://doi.org/10.1007/s11263-017-1059-x

[64] F. R. H. Zijlstra and L. Van Doorn. 1985. *The Construction of a Scale to Measure Perceived Effort*. University of Technology.

Received July 2019; revised January 2020; accepted February 2020