Explaining Reinforcement Learning to Mere Mortals: An Empirical Study

Andrew Anderson, **Jonathan Dodge**, **Amrita Sadarangani**, **Zoe Juozapaitis**, **Evan Newman**, **Jed Irvine**, **Souti Chattopadhyay**, **Alan Fern** and **Margaret Burnett**
Oregon State University
{anderan2, dodgej, sadarana, juozapaz, newmanev, irvine, chattops, Alan.Fern, burnett}@eecs.oregonstate.edu

Abstract

We present a user study to investigate the impact of explanations on non-experts' understanding of reinforcement learning (RL) agents. We investigate both a common RL visualization, saliency maps (the focus of attention), and a more recent explanation type, reward-decomposition bars (predictions of future types of rewards). We designed a 124 participant, four-treatment experiment to compare participants' mental models of an RL agent in a simple Real-Time Strategy (RTS) game. Our results show that the combination of both saliency and reward bars were needed to achieve a statistically significant improvement in mental model score over the control. In addition, our qualitative analysis of the data reveals a number of effects for further study.

1 Introduction

Although eXplainable Artificial Intelligence (XAI) has seen increasing interest as AI becomes more pervasive in society, much of XAI work does not attend to the *people* who consume explanations. In this paper, we draw upon a work that does, which introduced 4 principles for explaining AI systems to people who are not AI experts [Kulesza *et al.*, 2015]. These principles were: 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)." We ensured that our explanations were "sound": we did not approximate/simplify them. They were also "complete,": every agent input & output was represented in the UI.

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 [Kulesza *et al.*, 2015]. People's mental models, in the context of XAI, are basically their understanding of the way the agent works. More formally, mental models are, "*internal representations that people build based on their experiences in the real world.*" [Norman and Gentner, 1983]. 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.

In this paper, we investigate how people's mental models of a reinforcement-learning agent vary in response to different visual explanation styles–saliency maps showing where the agent is "looking," and reward decomposition bars showing the agent's current prediction of its future score. To do so, we conducted a controlled lab study with 124 participants across four treatments (saliency, rewards, both, neither), and measured both their understanding of the agent and their ability to predict its decisions. 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?
- **RQ-Predict** - Which treatment is better (and how) at enabling people to *predict* what the system will do?

2 Background & Related Work

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. While 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.

To address this problem, we draw on work by [Erwig *et al.*, 2018] for reward decomposition, which exploited the fact that rewards can typically be grouped into semantically meaningful types. 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 [Russell and Zimdars, 2003; Erwig *et al.*, 2018].

Before Erwig *et al.* (2018), others considered using reward decomposition [Russell and Zimdars, 2003; Van Seijen *et al.*, 2017]—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 trade-offs responsible for the agent's preference.

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 bar (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. [Simonyan *et al.*, 2013; Springenberg *et al.*, 2014; Zeiler and Fergus, 2014; Zhang *et al.*, 2018]). While there have been a number of comparison and utility studies (e.g. [Adebayo *et al.*, 2018; Ancona *et al.*, 2018; Greydanus *et al.*, 2018; Kim *et al.*, 2018; Riche *et al.*, 2013]), there is no consensus on a best way.

After exploring various techniques, we modified Fong and Vedaldi (2017)'s work on image classification, using a perturbation method that focused on *attributes* (blocks of pixels), rather than individual *pixels*, as used in computer vision, to aid human interpretation. Since the network may "focus" on different parts of the input for each reward bar, we compute saliency maps for each one, which the UI could visualize.

3 Methodology

We performed an in-lab study using a between-subjects design with **explanation style** (Control, Saliency, Rewards, Everything) as the independent variable. Our dependent variable was the quality of participants' mental models–measured by analysis of two main data sources: 1) answer to a post-task question, 2) accuracy of participants' prediction for the agent's selected action at each decision point (DP).

We ran an ablation study, where we measured the impact of each explanation by adding or removing them, as shown in Figure 1. Thus, Everything - Rewards - Saliency = Control, as follows: **1.** Control participants saw only the agent's actions, its consequences on the game state and the score (Region 1 & Figure 3), and question area (Region 4). **2.** Saliency participants saw Regions 1 & 4 plus Region 2, allowing them to infer intention from gaze [Newn *et al.*, 2016]. **3.** Rewards participants saw Region 1 & 4 plus Region 3, allowing them to see the agent's cost/benefit analysis. **4.** Everything participants saw all regions.

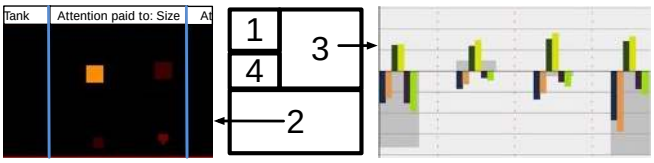

Figure 1: The regions of the interface. Region 1: game map, which we expand on in Figure 3. Region 2: saliency maps. Region 3: reward decomposition bars for each action. Region 4: participant question/response area. See text for who saw which regions.

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 & Human Sciences: 2 majors	5	1
Undisclosed	5	4
Totals	124	97

Table 1: Participant demographics, per academic discipline.

3.1 Participants And Procedures

With 2 ethics committees' approval, we ran 124 participants from 208 online 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 two-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). 78% of our participants, were "Gamers," defined as those with 10+ hours RTS experience, consistent with prior research [Penney *et al.*, 2018]. Afterwards, we noticed that gamers were spread evenly across treatments, so it was unnecessary to control for this statistically (Figure 2).

We began sessions with a 20-minute, hands-on tutorial to the *interface/domain*, with 3 practice DPs. Since participants were AI non-experts, we described saliency maps as, "*...like where the eyeballs of the AI fall*" and reward bars as, "*...the AI's prediction for the score it will receive in the future.*" Participants had 12 minutes to complete DP1 and 8 minutes per DP for the remaining 13.

The agent died 4 times; each time was a task boundary. Each task had 3-4 DPs, chosen for a mixture of diversity (e.g., all objects shown at least once), and similarity (e.g., some maps had the same object types but different health). At each DP, participants: (1) saw the game state with nothing else vis-

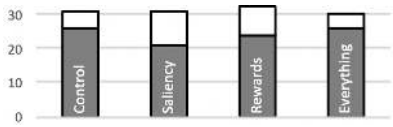

Figure 2: Distribution of RTS "gamers" in our study. "Gamers" are shown in grey, others in white.

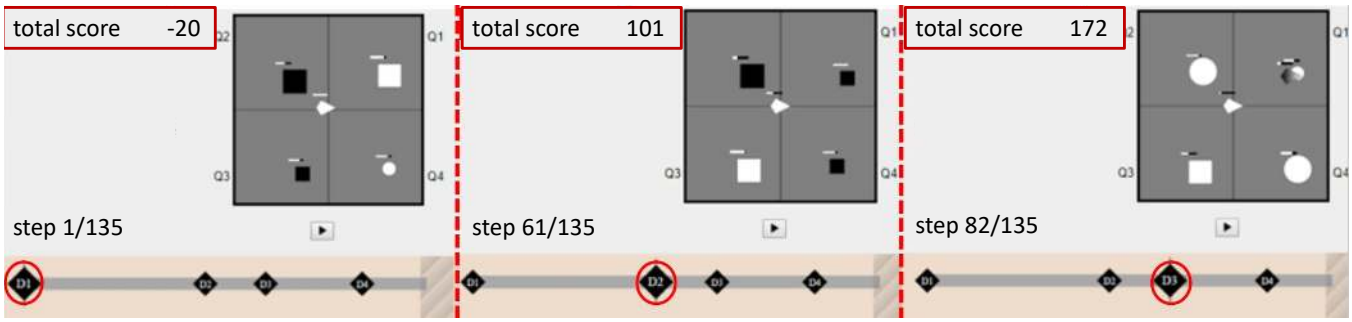

Figure 3: A sequence of the first three DPs of the game. For each DP (circled in red) participants saw the game map and the score (boxed in red). Next, they made a prediction of which object the AI would choose to attack. Last, they would receive an explanation and have the ability to "play" the DP. At DP1, the agent chose to attack Q2, causing a score change of 121 (+21 pts for damaging and +100 from destroying it).

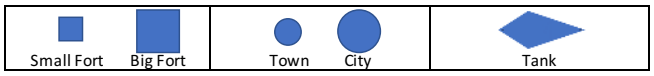

Figure 4: The objects appearing in our game states. Enemy objects were black, and allied objects were white.

ible yet; (2) answered questions about the object they thought the agent would attack and why; (3) upon submitting their answer, they saw what the agent did and the explanation for their treatment (reward bars, saliency maps, both, or neither). After all 14 DPs, they described the agent's decision making process, filled out a questionnaire, and received $20.

3.2 System Overview

Popularly available RTS games have an enormous action space – Vinyals *et al.* (2019) estimates $\approx 10^{26}$ for StarCraft II. With so many possibilities, it is not surprising that researchers have reported large differences in individual participants' focus, leading them to notice different decisions [Dodge *et al.*, 2018; Penney *et al.*, 2018]. To avoid this, we built a game with a tightly controlled action space to control the entire software stack (UI, agent API, etc). The game and study materials/code are here[1]

In our game, the agent's goal was to maximize its score over each task (Figure 3), subject to the following rules:

- Only Forts/Tanks could attack objects (Figure 4).
- At each DP, the agent *had to* attack one of the quadrants.
- If agent/friendlies were damaged/destroyed, it lost points.
- If enemies were damaged/destroyed, it gained points.
- Once the agent killed something, it "respawned" on a new map, carrying over its health.

The Reward Decomposition Implementation

The agent used six reward types to learn its $\vec{Q}(s,a)$: {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 $\vec{Q}(s,a)$. For each reward type c, there was a separate network for $Q_c(s,a)$ which took the state description as input–7, 40x40 greyscale image layers, each representing information about the state: {Health Points (HP), enemy tank, small forts, big forts, towns, cities, and friend/enemy}. The overall scalar Q-values $Q(s,a)$ used for action selection were the sum of each $Q_c(s,a)$. The agent trained using the decomposed SARSA learning algorithm using a discount factor of 0.9, a learning rate of 0.1, with ϵ-greedy exploration (ϵ decayed from 0.9 to 0.1). It trained for 30,000 games, at which point it demonstrated high-quality actions.

The Saliency Map Implementation

Given a state s, our perturbation-based approach produced a saliency map for each bar $Q_c(s,a)$ by giving data representing the true state s and a perturbed state s' (close to s), then subtracting the outputs for both states. Large output difference meant the system 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.* (2017) found it to be the most understandable for their participants. Our perturbations modify properties of *objects* in the game state and thus modify *groups* of pixels, not individual pixels.

Each of the perturbations represented a semantically meaningful operation: **1.** Tank Perturbation. If a tank was present, we removed it by zeroing out its pixels in the tank layer. **2.** Friend/Enemy Perturbation. Transform an object from *friend* to *enemy* by moving the friend layer pixels to the enemy layer (and vice versa). **3.** Size Perturbation. Transform an object from *big* to *small* (or vice versa) by moving the pixels from one size layer to the other. **4.** City/Fort perturbations. Similarly transform whether an object is a City or Fort. **5.** HP Perturbation. Since HP is real-valued it is treated differently. We perturbed the object HP values by a small value, 30%. These operations were represented in five saliency maps: HP, Tank, Size, City/Fort, & Friend/Enemy

To make the saliency maps comparable across types, we found the maximum saliency value in each map for *each* reward type & class from 16,855 episodes. Normalizing each map by this value placed the pixel value $\in [0,1]$.

4 Results: People Describing The AI

To elicit participants' understanding of the agent's decision making (RQ-Describe), we used Hsieh and Shannon (2005)'s summative content analysis on participants' answers to the end-of-session question: "*Please describe the steps in*

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

> 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)

Figure 5: Top scoring mental model question response. The highlighted portions illustrate both "basic" and "extra credit" concepts, some of which are described in Table 2.

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.*

Table 2: The four mental model codes revealing particularly interesting differences in nuances of participants' mental models.

the agent's approach or method..." [Hoffman *et al.*, 2018; Lippa *et al.*, 2008]. Figure 5 shows a sample response. Two researchers independently coded 20% of the data with 18 codes and reached $> 80\%$ agreement (Jaccard index) [Jaccard, 1908]. Given this reliability, one researcher did the rest.

In parallel with this process, we generated a rubric of scores to associate with each code. The codes representing the agent's four basic concepts were each worth 25% (e.g., its score maximization objective). Extra nuances in participants' descriptions earned small additions of "extra credit" (e.g., saying the agent maximized its *future* score), and extra errors earned small deductions (e.g., saying it tried to preserve its HP). Experimenting with different values for the small extras and errors had little effect on comparisons of score distributions among treatments. Figure 6 has the score distribution.

4.1 The More, The Better?

As Figure 6 shows, Everything participants had significantly better mental model scores than Control participants (ANOVA, F = 8.369, df = (1,59), p = .005[2]). One possible interpretation is that the Everything participants' performance was due to receiving the most sound and complete explanation, consistent with Kulesza *et al.* (2015)'s results.

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. Thus, we isolated each explanation type.

To isolate the effect of the reward bars, we compared all participants who saw the decomposed reward bars (the Rewards and Everything treatments) with those who did not. As the left side of Figure 7 illustrates, participants who saw reward bars had significantly better mental model scores than those who did not (ANOVA, F = 6.454, df = (1,122), p = .0123). Interestingly, isolating the effect of saliency produced a similar impact. As the right side of Figure 7 illustrates, those who saw saliency maps (the Everything + Saliency treatments) had somewhat better mental model scores (ANOVA, F = 3.001, df = (1,122), p = .0858). This suggests that each component brought its own strengths.

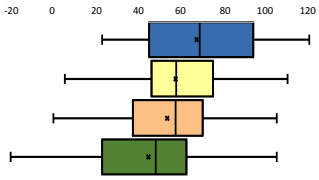

Figure 6: The participants' final mental model scores. Box colors from top to bottom: Everything, Rewards, Saliency, and Control.

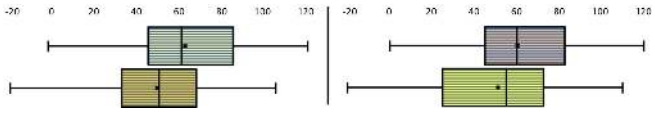

Figure 7: Same data as Figure 6. Left: Mental model scores for those who saw *rewards* (top) and those who did not. Right: Same data, but those who saw *saliency* (top) and those who did not.

4.2 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 2 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 2's Maximize Score). Over 3/4 (36 out of 46) of the people who mentioned this were in treatments that saw rewards. For example: "*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.*" (R81)[3] and: "*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.*" (E14)

Some participants who saw rewards also mentioned the nuance that the agent's interest was in its *future* score (Table 2's Forward Looking), not the present one: "*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...*" (E83). Over 2/3 of the participants (9 out of 13) who pointed out this nuance saw decomposed reward bars.

Even more subtle was the agent's paranoia (Table 2's Paranoia). It had learned Q-value components that reflected a

[2] We consider $p < .05$ significant, and $.05 \leq p < .1$ marginally significant, by convention [Cramer and Howitt, 2004].

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

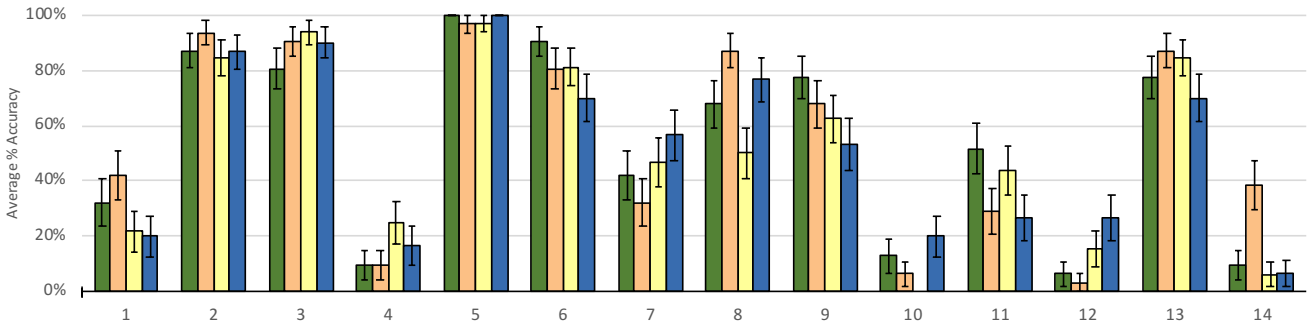

Figure 8: Percentage of participants who successfully predicted the AI's next move at each decision point (DP). Bar colors denote treatment (from left to right): Control, Saliency, Rewards, and Everything. Participants' results varied markedly for the different situations these DPs captured, and there is no evidence that any of the treatments got better over time. All error bars ($SE = \sigma/\sqrt{n}$) are under 10%.

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. After investigating, we determined that this was a result of learning via the on-policy SARSA algorithm[4], which learns while it explores.

This paranoia can be a type of "bug" in the agents value estimates. The only 8 participants in the entire study who pointed out this bug were participants who saw rewards. For example: "*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.*" (R73).

On the other hand, 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 2/3 (10 out of 15) of the participants who talked about such behaviors were those who saw saliency maps. As one participant put it: "*If it cannot take down any structures, it will throw itself to wherever it thinks it will deal the most damage.*" (S74).

4.3 Discussion: Which Explanation?

On the surface, Section 4.1 suggests that, in explainable systems, the more explanation we give people, the better. However, Section 4.2 suggests that the question of which explanation or combination of explanations is better is more complex – each type has different strengths, which may matter differently in different situations. To investigate how situational an explanation type's strength is, we turn next to a qualitative view of how participants fared in individual situations, which we captured with their predictions at each DP.

[4]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.

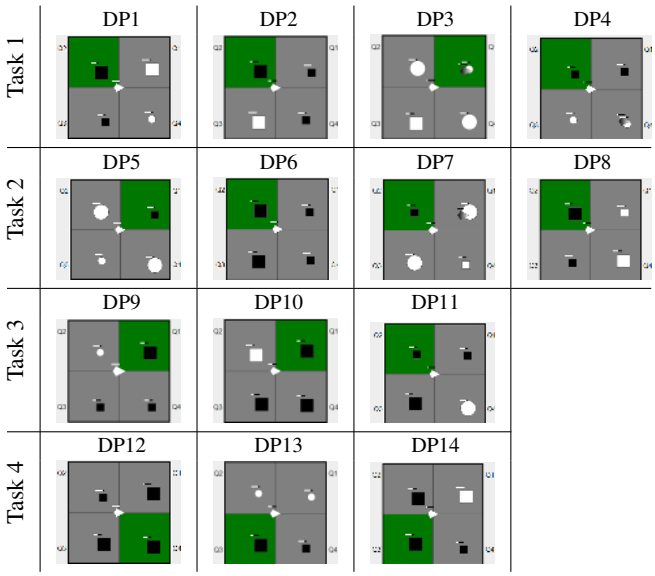

Table 3: The tasks and their DPs. We have highlighted in green the action the AI chose.

5 Results: People Predicting The AI

Participants' action predictions provided us *in situ* data [Hoffman *et al.*, 2018; Muramatsu and Pratt, 2001]. Recall that at each DP (Table 3), participants did not see explanations until they predict the agent's action. As Figure 8 shows, participants' accuracy varied and showed no learning effect. To understand why, we used qualitative analysis to investigate further, as we detail next.

5.1 Help! The Choice Is Counter-Intuitive

Situations where the agent went against participants' intuitions proved confusing. These cases all had low accuracy, with *all* treatments' below random guessing ($\leq 25\%$).

One of these situations was the agent choosing neither the strongest nor weakest of similar enemies (DPs 10,12). When the Everything participants got it right, their comments suggest they combined both saliency and rewards into their reasoning; e.g., (E71) for DP10: "*As it will **look at the HP** of*

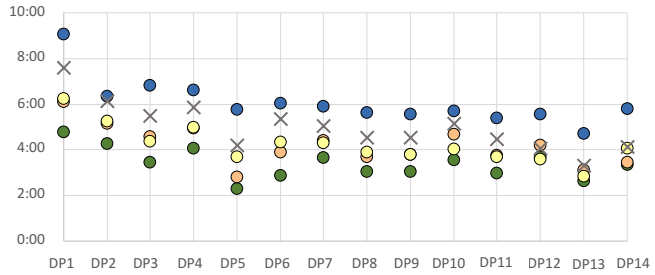

Figure 9: Average task time vs DP, per treatment. Participants had 12 minutes for the first DP and 8 minutes for all subsequent DPs. "×": see text. Colors: same as Figure 8

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.*"

However, *all* of the participants in the Rewards treatment got DP10 wrong, suggesting the need for saliency maps to show how much the AI focused on its *own* HP. For example: "*Lowest HP out of the 3 big fort.*" (R94).

A second situation that was counter-intuitive to participants was the agent choosing to attack an enemy elsewhere over saving a friend. The worst accuracy for this type was at DP4: 77% of the participants got it wrong. They incorrectly predicted it would attack the enemy tank, citing its health: "*This is the enemy object with the lowest value for HP.*" (S18) or its threat to a friend "*The enemy tank poses the greatest threat [to] allies...*" (S25). Of the few participants (19 total) who predicted correctly, most (68%) were in treatments that saw reward bars, e.g.: "*... destroying enemy [fort] will give you more point than destroying a tank.*" (R94).

5.2 Overwhelmed!

In DPs 6, 9, & 11, the Everything participants' had the lowest predictive accuracy, while Control had the highest. This seems tied to Everything participants coping with too much information, showing the importance of balancing completeness without overwhelming users [Kulesza *et al.*, 2015].

Some Everything participants tried to account for *all* the information they had seen. For example, at DP6: "*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.*" (E38). Some bemoaned the complexity of the information: "*It was confusing all around to figure out the main factor for movement using the maps and bars...*" (E39). In contrast, participants in the Control were able to apply simpler reasoning for the correct Q2 prediction at DP6: "*because it is the lowest health of all of the enemy objects.*" (C69).

Figure 9 attests to Everything participants' burden of processing all the information. In the figure, "×" 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. Everything participants' time to act upon their explanations exceeded the summed time of the parts, at *every* DP. Further, since we limited DP time, some "timed out"–and Everything accounted for 63% of them.

5.3 No Help Needed... Yet

For some DPs (2, 3, 5, 13), explanations seemed unnecessary, as the Control proved "good enough" (at least 75% of participants predicted correctly). At "Easy" situations, explanations may simply interfere. However, it may not be easy for everyone. On-demand explanations can provide more information to those who need it, without overwhelming those that do not.

5.4 Discussion: Its All Depends...

Participants' explanation needs depended on the situation; hence the variability illustrated in Figure 3. Statistically, treating these situations together simply "cancels out" effects. This wide variation should be expected, given the variability in state/action pairs, combined with the noisiness of *human data*. The mix of quantitative with qualitative methods for RQ-Predict served us well, and we recommend it to other XAI researchers facing similarly situation-dependent data.

6 Threats To Validity

Any study has threats to validity [Wohlin *et al.*, 2000]. In our study, participants' proficiency in RTS games might have helped understand the agent's tactics. Although the RTS "gamers" were fairly evenly distributed across our treatments (Figure 2), we did not collect many demographics, preventing us from 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, to reduce confounding factors, we simplified our game, but this means that our results might not hold for complex RTS games. We controlled every participants' time per DP, but this may have impacted their mental models with: insufficient time per DP (8 minutes) and also too few DPs (14). Threats like these can only be addressed with more studies using diverse empirical methods to generalize the findings.

7 Concluding Remarks

In this paper, we report on a mixed methods study with 124 AI non-experts. We investigated which of four visual explanation possibilities–saliency, rewards, both, or neither–helped them build the most accurate mental models, and in what circumstances. Among the surprising results were:

- Everything participants had significantly better mental model description over the Control participants (Section 4).
- Rewards participants had the most insight into nuanced concepts, such as agent paranoia (Section 4.2).
- Participants needed entirely different types of explanations for different situations (Section 5).
- We corroborated Kulesza *et al.* (2015)'s results about not overwhelming users in a new domain (Section 5.2).

Our analyses suggest several one-size-does-*not*-fit-all takeaway messages. First, one type of explanation does not fit all *situations* (Section 5). Second, one type does not fit all *people* (Figure 6). And perhaps most critical, one type of empirical analysis (only quantitative or qualitative) was not enough; we needed both to make sense of the variation between individual participants and individual DPs. We believe that, only by our community using an arsenal of empirical techniques, can we hope to learn how to explain AI effectively to mere mortals.

References

[Adebayo *et al.*, 2018] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems 31*, 2018.

[Ancona *et al.*, 2018] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *Intl. Conf. on Learning Representations*, 2018.

[Cramer and Howitt, 2004] Duncan Cramer and Dennis Howitt. *The Sage dictionary of statistics: a practical resource for students in the social sciences*. Sage, 2004.

[Dodge *et al.*, 2018] Jonathan Dodge, Sean Penney, Claudia Hilderbrand, Andrew Anderson, and Margaret Burnett. How the experts do it: Assessing and explaining agent behaviors in real-time strategy games. In *ACM Conf. on Human Factors in Computing Systems*, CHI '18. ACM, 2018.

[Erwig *et al.*, 2018] Martin Erwig, Alan Fern, Magesh Murali, and Anurag Koul. Explaining deep adaptive programs via reward decomposition. In *IJCAI Workshop on Explainable Artificial Intelligence*, 2018.

[Fong and Vedaldi, 2017] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[Greydanus *et al.*, 2018] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding Atari agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*. PMLR, 2018.

[Hoffman *et al.*, 2018] Robert Hoffman, Shane Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv:1812.04608*, 2018.

[Hsieh and Shannon, 2005] Hsiu-Fang Hsieh and Sarah Shannon. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9), 2005.

[Jaccard, 1908] Paul Jaccard. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.*, 44, 1908.

[Kim *et al.*, 2018] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Intl. Conf. on Machine Learning*, volume 80. PMLR, 2018.

[Kulesza *et al.*, 2015] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *ACM Intl. Conf. on Intelligent User Interfaces*, IUI '15. ACM, 2015.

[Lippa *et al.*, 2008] Katherine Lippa, Helen Klein, and Valerie Shalin. Everyday expertise: cognitive demands in diabetes self-management. *Human Factors*, 50(1), 2008.

[Muramatsu and Pratt, 2001] Jack Muramatsu and Wanda Pratt. Transparent queries: investigation users' mental models of search engines. In *Intl. ACM SIGIR Conf. on Research and Development in Info. Retrieval*. ACM, 2001.

[Newn *et al.*, 2016] Joshua Newn, Eduardo Velloso, Marcus Carter, and Frank Vetere. Exploring the effects of gaze awareness on multiplayer gameplay. In *ACM Symp. on Computer-Human Interaction in Play Companion Extended Abstracts*. ACM, 2016.

[Newn *et al.*, 2017] Joshua Newn, Eduardo Velloso, Fraser Allison, Yomna Abdelrahman, and Frank Vetere. Evaluating real-time gaze representations to infer intentions in competitive turn-based strategy games. In *ACM Symp. on Computer-Human Interaction in Play*. ACM, 2017.

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

[Penney *et al.*, 2018] Sean Penney, Jonathan Dodge, Claudia Hilderbrand, Andrew Anderson, Logan Simpson, and Margaret Burnett. Toward foraging for understanding of StarCraft agents: An empirical study. In *ACM Intl. Conf. on Intelligent User Interfaces*, IUI '18. ACM, 2018.

[Riche *et al.*, 2013] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: state-of-the-art and study of comparison metrics. In *IEEE Intl. Conf. on Computer Vision*, 2013.

[Russell and Zimdars, 2003] Stuart Russell and Andrew Zimdars. Q-decomposition for reinforcement learning agents. In *Intl. Conf. on Machine Learning*, 2003.

[Simonyan *et al.*, 2013] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, 2013.

[Springenberg *et al.*, 2014] Jost Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, 2014.

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

[Vinyals *et al.*, 2019] Oriol Vinyals, David Silver, et al. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/, 2019.

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

[Zeiler and Fergus, 2014] Matthew Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conf. on Computer Vision*. Springer, 2014.

[Zhang *et al.*, 2018] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *Int. J. Comput. Vision*, 126(10), October 2018.