How Do People Rank Multiple Mutant Agents?

Jonathan Dodge
Andrew Anderson
Matthew Olson
Rupika Dikkala
Margaret Burnett
dodgej@eecs.oregonstate.edu
anderan2@eecs.oregonstate.edu
olsomatt@oregonstate.edu
dikkalar@oregonstate.edu
burnett@eecs.oregonstate.edu
Oregon State University
Corvallis, OR, USA

ABSTRACT

Faced with several AI-powered sequential decision-making systems, how might someone choose on which to rely? For example, imagine car buyer Blair shopping for a self-driving car, or developer Dillon trying to choose an appropriate ML model to use in their application. Their first choice might be infeasible (i.e., too expensive in money or execution time), so they may need to select their second or third choice. To address this question, this paper presents: 1) Explanation Resolution, a quantifiable direct measurement concept; 2) a new XAI empirical task to measure explanations: "the Ranking Task"; and 3) a new strategy for inducing *controllable* agent variations—Mutant Agent Generation. In support of those main contributions, it also presents 4) novel explanations for sequential decision-making agents; 5) an adaptation to the AAR/AI assessment process; and 6) a qualitative study around these devices with 10 participants to investigate how they performed the Ranking Task on our mutant agents, using our explanations, and structured by AAR/AI. From an XAI researcher perspective, just as mutation testing can be applied to any code, mutant agent generation can be applied to essentially any neural network for which one wants to evaluate an assessment process or explanation type. As to an XAI user's perspective, the participants ranked the agents well overall, but showed the importance of high explanation resolution for close differences between agents. The participants also revealed the importance of supporting a wide diversity of explanation diets and agent "test selection" strategies.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in HCI**.

KEYWORDS

Explainable AI, After-Action Review

ACM Reference Format:
Jonathan Dodge, Andrew Anderson, Matthew Olson, Rupika Dikkala, and Margaret Burnett. 2022. How Do People Rank Multiple Mutant Agents?. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3490099.3511115

1 INTRODUCTION

Explaining episodic decisions is a significant challenge with much ongoing XAI research (e.g., [9, 21]). Explaining decisions in *sequential* domains is even more challenging, as decisions must be explained in relationship to previous ones (and possible future ones). Still more challenging is explaining decisions in sequential domains with the goal of enabling users to go beyond picking the "best" agent—to *partially ordering* agents with respect to some (set of) properties.

For example, imagine car buyer Blair trying to select among self-driving cars. Blair may perceive the best performer as too expensive, delayed, etc. Similarly, developer Dillon may decide among off-the-shelf ML models to incorporate into an application, e.g., as described in Hill et al. [31]. Dillon might not use a standard benchmark-leading model because its API may be intimidating, underlying model difficult to comprehend, or execution too costly. Thus, while Blair and Dillon may not need to *fully order* all candidates, they might create a shortlist for some, in case they have to resort to their second or third choice. Such a ranking process requires the assessor to consider and compare available agents, thinking critically about their strengths and weaknesses.

Researchers have investigated a wide variety of stakeholders who might be assessing the properties of AI [17, 44]. In fact, as the ACM code of ethics points out in item 1.1, *"...all people are stakeholders in computing."* Although not *every* human is necessarily in-the-loop, anyone *could* be. One way to support diverse humans assessing AI is via explanation. As Hoffman et al. write: *"By hypothesis, explanations that are good and are satisfying to users enable users to develop a good mental model."* [32].

IUI '22, March 22–25, 2022, Helsinki, Finland

ACM ISBN 978-1-4503-9144-3/22/03…$15.00
https://doi.org/10.1145/3490099.3511115

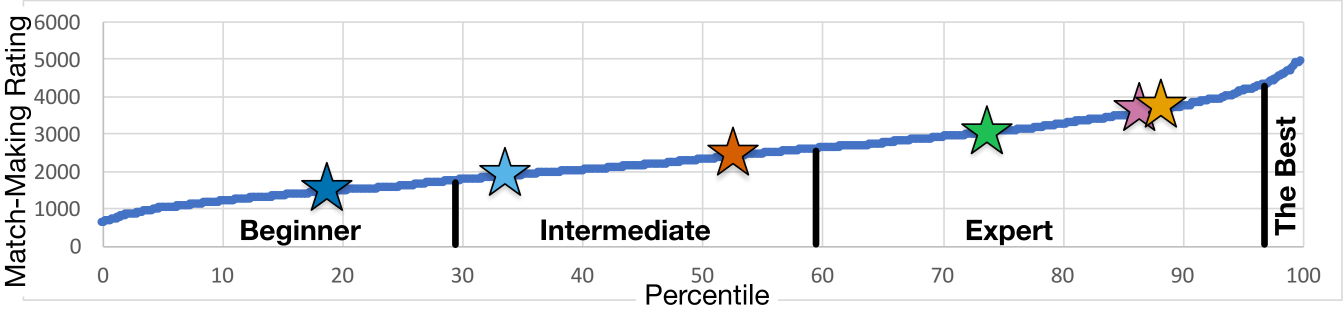


Figure 1: Hypothetical Matchmaking Rating (MMR) chart in a game showing the distribution of players' skill, akin to figures from Vinyals et al. [96] or Robertson et al. [77]. The background line is the whole player population, and the stars correspond to the true skill levels of a collection of agents to assess. It may be possible to differentiate the MMR property of an expert from a beginner (Orange and *Blue*) simply from watching them once because of the large gap. Meanwhile, resolving the difference between two experts (Orange and *Green*) is much more difficult, and may require explanation. As the agents become more similar (Orange and *Pink*), they may become impossible for humans to rank, even with explanations.

Given an explanation, is ranking agents for such purposes viable for humans like Blair and Dillon? The answer may depend on a property we term *explanation resolution*, based on microscopy's concept of resolution, defined[1] as *"the shortest distance between two points on a specimen that can still be distinguished by the observer...as separate entities"*. Thus, an explanation with *high* resolution should enable an observer to distinguish not only agents that greatly differ (e.g., Beginner vs. Expert in Figure 1), but also agents that do not (e.g., the two top agents)—ideally in a *prospective* fashion, before large amounts of performance data are available.

We propose that explanation resolution can be empirically measured—and that doing so enables evaluation of explanations' support of use-cases such as Blair's and Dillon's. More generally, measuring explanation resolution also enables XAI researchers to empirically *compare* alternative explanation strategies on the basis of how well each can reveal differences among agents.

To measure explanation resolution empirically requires a suitable empirical task. To that end, this paper first introduces the *Ranking Task*, an XAI empirical assessment task for use-cases involving humans doing (partial) ordering. Using this task, XAI empiricists can measure participants' efficacy in ranking the agents with scoring mechanisms such as how many agents a participant ranked exactly correctly and how "far off" a participant's ranking of an agent was from its true rank.

Ranking occurs with respect to one or more properties, such as winningness, fairness, etc. If the desired property can be *manipulated* in a controllable fashion, XAI empiricists can cleanly measure the quality of an explanation and/or assessment process by its ability to expose that such manipulation has occurred. To address this need, we introduce *Mutant Agent Generation*, a manipulation approach inspired by software engineering's comparisons of different test suites via Mutation Testing [8, 16, 68].

Using the Ranking Task and Mutant Agent Generation, we conducted a qualitative study to investigate how participants would rank six sequential decision-making agent mutations playing MNK games (a generalization of Tic-Tac-Toe). To support participants' ranking, we created three novel explanations designed for use in sequential domains. To loosely structure the participants' investigation, we adapted a process called After-Action Review for AI (AAR/AI) [20, 42, 55] to support assessment at the granularity of games. Our aim was to glean from participants their efficacy at the Ranking Task; which explanations they relied upon to perform it; how they went about comparatively assessing the different agents; and how they invested their limited time to perform these comparisons.

This paper offers the following main contributions:

(1) Explanation Resolution, a quantifiable direct measurement concept;
(2) Ranking Task, enabling XAI researchers to measure explanation resolution; and
(3) Mutant Agent Generation, allowing controllable variation among agents for systematic empirical investigation.

In support of the main contributions, we also contribute:

(4) Novel explanations supporting sequential decision-making agents;
(5) Adaptation of AAR/AI to a higher level of granularity (games instead of decisions); and
(6) Qualitative empirical investigation into how participants performed the Ranking Task on agents produced via Mutant Agent Generation, using the explanations we provided and scaffolded by AAR/AI.

2 BACKGROUND

2.1 Explanations and Users' Mental Models

Hoffman et al. hypothesized that a *"good mental model will enable [users] to develop appropriate trust in the AI..."* [32]. Those authors enumerated a number of mental model elicitation strategies (detailed in [32]'s Table 4). Among them, many are qualitative, (e.g., Think Aloud or Interview techniques), while others are more quantitative (e.g., Retrospection or Prediction Tasks).

However, users have little foundation on which to build a mental model if unable to inspect system behavior. Many ML systems appear as opaque boxes, with little explanation as to *why* the system

[1] https://www.microscopyu.com/microscopy-basics/resolution

provides outputs [48]. The role of explanations is to make such boxes less opaque. Explanations have been shown to improve mental models [48, 49], satisfaction (in the colloquial sense, the user's self-reported feeling) [2, 40], and understanding—particularly in low expertise observers [103]. Still other research shows that explanations are not necessarily a panacea. For example, some research showed less dramatic effects, as the overall structure of participant mental models went largely unchanged, though it did seem to help dispel misconceptions [92].

In order to inform a mental model given potentially complex explanations, including charts and figures, participants may support acquiring transferable knowledge [75] via "self-explanations." In particular, learners employing less successful strategies rely heavily on examples, struggling to self-explain [13]. However, participants' willingness to engage with the explanation will be moderated by individual differences—which makes measuring such differences important. For example, research shows that some prefer superficial explanations, while others prefer explanations that support more deliberative reasoning [23, 45].

2.2 Explaining in Sequential Domains

Most work in *XAI* does not focus on sequential domains, a gap which leaves extensive work by *AI* researchers largely unharnessed. For example, AI researchers have long studied domains like Chess, Shougi, and Go, recently reaching performance exceeding the best humans [85, 86]. As Reeves et al. [74] put it, *"Game expertise... is constantly concerned with 'why that now,' 'where can I go from here,' 'what next,'... familiar concerns for those who study the sequential ordering of human action."* In Real-Time Strategy (RTS) games, DeepMind's AlphaStar agent has achieved sustained top notch performance over a whole ladder season deployed with humans [96, 97]. Explaining RTS actions remains a challenge, though Metoyer et al. [57] studied how expert-novice pairs do it. More recently, Penney et al. [67] examined how professional commentators and lab participants behaved in an effort to inform explanation design. For example, Madumal et al. [54] created RTS explanations by extracting paths from a causality graph.

Selecting action/states to observe is a specific challenge for sequential domains. Hayes and Shah applied predicates to a set of states, succinctly summarizing a mapping to that set [29]. Applying predicates at the trajectory level (as opposed to *state*) can help group low level actions into more abstract subtasks (e.g. a car "changing lanes") [35]. Another approach from Huang et al. [34] seeks to select states via *criticality* (max_action - average_action). Amir and Amir [4] offer a different name (*importance*) for a slightly different function (max - *min*).

Summarizing the policy globally poses unique challenges in sequential domains. Zahavy et al. [105] used a large t-SNE plot to navigate the state space. Another strategy, modifying reward functions to train modified policies allows predicate testing to explain actions [94]. Olson et al. [62] analyzed policy trajectories by generating counterfactuals for critical states. Other promising strategies explaining policies globally: extracting automata [100], rules [60], or decision trees [106].

2.3 "Testing" AI

AI and ML systems have some important terminological differences from traditional software systems that bear clarifying. For example, Groce et al. [26] argue that an AI system is itself a "program", but with no source code. The learned program may have come from a flawless AI algorithm, but the learning process could still introduce faults, e.g., from biased training data. Those authors write that the meaning of identifying and correcting a fault in such a source-less program, *"must be parametrized with respect to the fault-correction method(s) available."* [26]. Thus, traditional software interventions like fixing a line of code are not necessarily meaningful; instead ML/AI systems offer different correction techniques. For example, Goodfellow et al.'s Chapter 11 [25] recommends: *"Visualize the model in action, Visualize the worst mistakes, Reason about software using training and test error, Fit a tiny dataset, Compare back-propagated derivatives to numerical derivatives, and Monitor histograms of activations and gradient."* [25]. However, this list of interventions assumes the assessor has deep ML/AI knowledge.

In light of the uniqueness of the machine learning pipeline [31], some analysis tools allow the user to inspect each element of the pipeline. One of these, GAN Lab by Kahng et al. [39] is intended for instructional purposes. Another, ModelTracker by Amershi et al. [3], supports debugging by inspecting system performance at the example level. Conversely, some tools treat the system as opaque, splitting outputs into groups and comparing performance between the different cohorts [38]. Other techniques that operate on opaque boxes include: LIME [76], inspection of predictions [46, 53], test selection [26, 83], and counterfactuals [98].

Interactive tools are a very recent alternative to opaque box explanations. Though they are often published per information visualization literature, they offer some of the more powerful mechanisms for inspecting complicated ML systems. Some are for instructional purposes, (e.g. [39]), but others are state-of-the-art systems [11, 37, 63]. More recently, these techniques have been applied to models for data scientists [33].

Neural networks are among the hardest systems to test, recent work shows that modern networks still succumb to relatively simple image manipulations, meant to mimic graffiti on stop signs [22]. Carlini et al. also demonstrated adversarial examples which can cause a *"network to incorrectly classify images by changing only the lowest order bit of each pixel."* [10]. To address these challenges, researchers have devised a number of strategies, surveyed by Gilpin et al. [24]. One is rendering the network more interpretable while attempting to maintain performance [12, 51]. Another is to *verify* networks, surveyed by Melis et al. [56]. One of the most recent describes DeepTest [90], which attempts to use image processing for data augmentations. These extra images achieve greater "neuron coverage" [66], which is analogous to "code coverage" from software engineering.

2.4 Humans Assessing AI, Qualitatively

In software engineering, a complement to testing is code inspection: qualitatively checking the system's reasoning to find flaws. As Kulesza et al. pointed out, an analogous AI inspection approach would check the explanations themselves [48, 50].

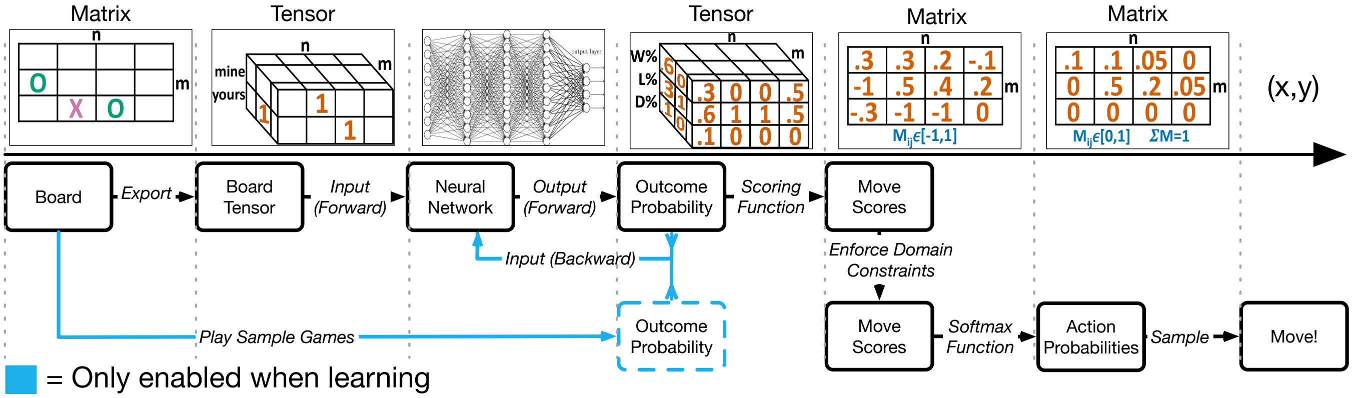


Figure 2: How the agent makes a decision, showing nouns in black boxes, verbs on arrows, and the data involved above each step in the pipeline. The process begins with a board, which gets converted to a board tensor, then passed into the CNN. The CNN outputs an outcome probability tensor, which is then scored, resulting in a position score matrix. After enforcing domain constraints on the score matrix, we softmax the scores, and sample from the resulting distribution to select a position. Parts in cyan activate only during training.

To support humans assessing AI, some argue for the importance of systematic *process*. Toward that end, some researchers have turned to techniques humans use to assess *humans*. After-Action Review (AAR) is one of several such process-oriented approaches for human assessment of humans [81]. The AAR was devised by the U.S. Army [59, 93], but sees continued use [7, 28, 78]. AAR has been adapted for use in other domains, such as medical treatment [79], emergency preparedness [15], fire fighting [36] and AI via a variant known as AAR/AI [20, 55].

The AAR/AI process works by taking the human assessor through a range of assessment perspectives, like: *"what happened?"*, *"why?"*, and *"how can we do better?"*. It does so in a loop, starting with set-up and concluding with learning formalization, specifically the steps are [20, 55]: Define rules, Explain objectives, Review what was supposed to happen, Identify what happened, Examine why, Formalize learning, and Formalize learning from the whole session. Khanna et al. [42] showed that human participants assessed more effectively when using AAR/AI with explanations, compared to when using the same explanations without the AAR/AI process. Specifically, they observed that participants using AAR/AI found *more* bugs, with *higher* precision than their counterparts who did not. This result is consistent with a meta-analysis of AAR-based methods, observing (on average) a large practical effect [41]. Our experiment includes an adaptation of AAR/AI.

3 THE EXPLANATIONS; AND THE AGENTS THAT GENERATE THEM

Section 2 pointed to significant work to explain an AI agent, but few such explanations are aimed at comparing multiple agents' logic. In this section, we describe the three interactive explanations we created to empower participants to comparatively assess our agents, which work as described in Figure 2.

Figure 3 will show the explanations' environment[2]. The control panel is on the left, the game board at the top right, and the explanation just below the game board. This arrangement of explanation and state juxtaposes them—making our designs more generalizable, though possibly harder to use than if we had superimposed them [14]. The three explanations appear in Figures 3, 4, and 5; each with the same board state and mouse position (yellow highlighted square in Figure 3).

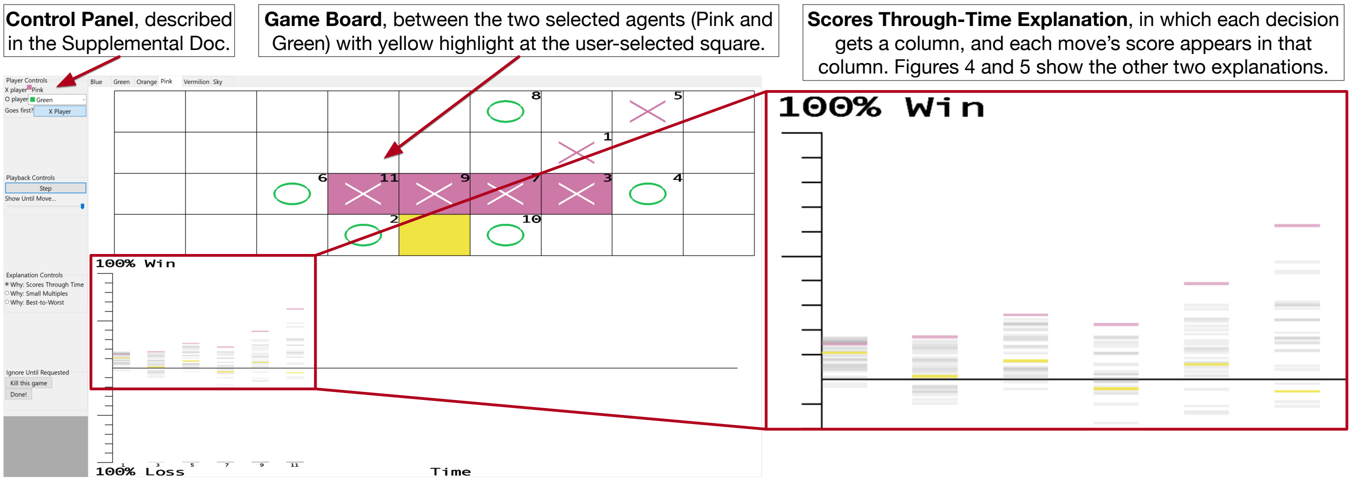
3.1 The Agent

The agent generating these explanations has a convolutional neural network tasked with predicting outcome tuples $O = (Win\%, Loss\%, Draw\%)$ for *each* square, given only the $M \times N$ board (Figure 2). The network has an input layer with 2 channels (the agent's pieces are always in channel one, the opponent's in channel two). Thus, the network input tensor has dimension $M \times N \times 2$.

Provided for context and never visible in the interface, the agents' internal structure was: The input tensor goes through 5 convolutional layers, each with kernel size 3 (only the third layer uses a stride, set at 2), followed by 2 fully connected layers, and ending with a sigmoid layer. Next, the fully connected linear layers compress the tensor to a vector of length $M \times N$, which is expanded to its final shape. The output shape is $(M \times N \times O)$, where O is the set of outcomes—in this case a 3-element vector. The network uses the ReLU activation function throughout.

To select actions, the agent starts with a forward pass on the network. Then, armed with predicted outcomes from the network, the agent uses them in a generalized value function (proposed by Sutton et al. [87], though still used, e.g., [52]). Our agent's scoring function is defined as: $(Win\% - Loss\%)$. Last, it applies a Gumbel-softmax function to the scores, before sampling from the resulting distribution to pick a *near*-max valued action. This softmax has a

[2] Our program is written in Python, with dependencies for GUI elements (wxPython [88]), graphics (OpenGL [84]), and neural networks (PyTorch [65]). Full implementation: https://github.com/dodgej/RankingMutants



Control Panel, described in the Supplemental Doc.

Game Board, between the two selected agents (Pink and Green) with yellow highlight at the user-selected square.

Scores Through-Time Explanation, in which each decision gets a column, and each move's score appears in that column. Figures 4 and 5 show the other two explanations.

Player Controls

X player Pink

O player 🟩 Green

Goes first? X Player

Playback Controls

Step

Show Until Move...

Explanation Controls

● Why: Scores Through Time

○ Why: Small Multiples

○ Why: Best-to-Worst

Ignore Until Requested

Kill this game

Done!

Figure 3: The environment, showing the Scores Through-Time explanation (Section 3.2). The control panel (left) allowed participants to pick agents, an explanation type to view, and step through the games The game board (top) shows where X (pink) and O (green) moved, and the highlighted X's show that X got 4-in-a-row, thereby winning the game. The Scores Through-Time explanation (callout, right) answers the question "at each step, how did the pink (X) agent score each move? The one it chose (pink) at each step is always near the top-scorer. The user's cursor is on the yellow-highlighted game board square, which similarly highlights the scores corresponding to that move in the explanation. Figures 4 and 5 show the other two explanations.

temperature parameter, used to mediate the explore/exploit tradeoff during *training*. Afterwards, we maintain 0.1 temperature so the agents encode a probabilistic policy, and games are not deterministic given a pair of agents. Consult our Supplemental Documents for more details.

For the backward pass, we compute L1Loss between the network's output and the target values. To compute targets, we do uniform random sampling on decisions available at the current state, with 10 rollouts per decision to estimate value. We then compute proportions of win, loss, and draw from the results of the game rollouts—these values become regression targets. This formulation makes the learning problem difficult, but provides explanation information about *all* decisions with a single forward pass, making our agent easily run at interactive rates on consumer hardware.

3.2 Explanation 1: Scores Through-Time (*StTime*)

This explanation emphasizes the *time* dimension of the data, attempting to answer: "At each *decision*, how did the agent score each square?"

The Scores Through-Time (*StTime*) explanation uses time as the X-axis, and whenever the agent being assessed makes a decision, a new column appears with the agent's scoring of every potential square at that decision. For example, at decision 11 in Figure 3 the Pink X player's highest-scoring square was also the one it took (in pink).

In each column, one rectangle is the same color as the agent (Pink) which depicts the agent's scoring of the square it selected. Other rectangles show the agent's scorings of the 35 not-selected squares on the 36-square board, including illegal decisions. (If the agent is well trained, the illegal decisions are assigned very low scores.) Hovering over any gameboard square highlights its scoring for every decision through time. Hovering over any scoring in the explanation highlights the squares on the gameboard associated with that score. If the participant moves the game forward one step, the explanation adds a new column for that decision point.

3.3 Explanation 2: Scores On-the-Board (*OnBoard*)

This explanation emphasizes the relationship between the *time* and *space* dimensions of the data, at the cost of adding complexity from using multiple charts. Thus, it attempts to answer: "How good is this *move* at different points in time?"

The Scores On-the-Board (*OnBoard*) explanation divides the *StTime* explanation to give each decision its own chart instead of combining all decisions into one chart. Each grid element is an *StTime* chart for one decision, intended to ease comparison [91]. For example, the top left chart in Figure 4 shows the agent's perception of its likelihood of winning if it placed an X in the top left square at any prior decision.

Each chart contains its own *StTime* explanation for that square, with the same coordinate axes and decision number labels. Thus, the far left of each chart represents the score X gave that particular square at the first decision.

Meanwhile, the far right represents the score X gave to the same square at the most recent decision. Hovering on any gameboard square highlights the explanation corresponding to that square. Hovering over any of the decision's explanations highlights the gameboard square—clarifying the spatial alignment of the grids of pieces and charts. Moving the game forward one step adds one new data point to each chart for the most recent decision.

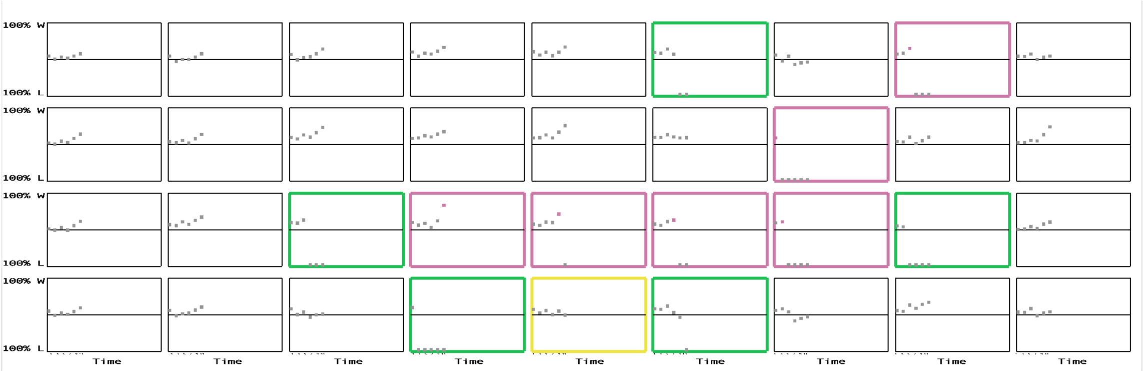


Figure 4: The Scores On-the-Board Explanation. Each move gets a small chart of Scores Through-Time, with occupied squares colored by the agent's color (pink and green; yellow indicates the square highlighted in Figure 3). Figure 3 uses our old name for this explanation.

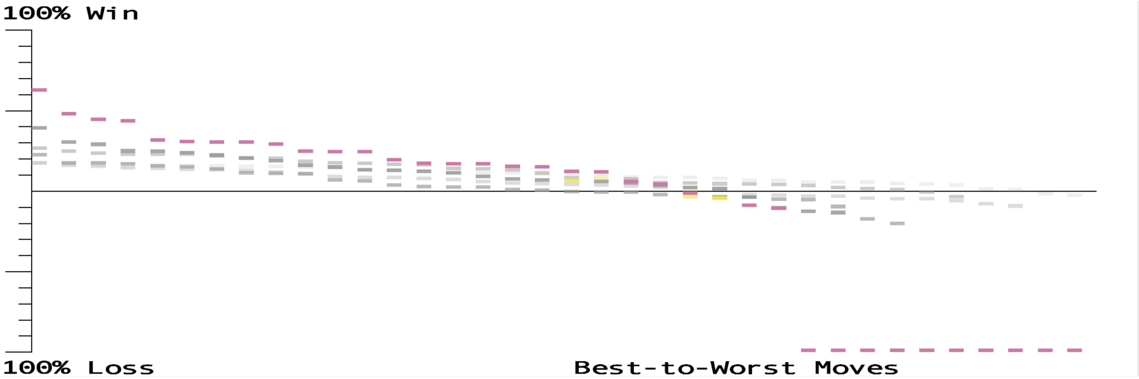


Figure 5: The Scores Best-to-Worst Explanation. Each decision results in a single sorted data series, which are identified by color (pink is the most recent, then grey colors from dark to light).

3.4 Explanation 3: Scores Best-to-Worst (*BtoW*)

This explanation emphasizes the *value* dimension of the data, attempting to answer: "At each decision, how did the agent score each *square*?"

The Scores Best-to-Worst (*BtoW*) explanation reframes the focus—onto options (game squares) instead of time. Given that at each decision, selecting an action requires considering multiple actions, and a single episode contains multiple decisions, storing all the values results in a tensor with both space dimension and time dimension. *BtoW* explanation cuts along the space dimension.

BtoW has the same Y-axis as the other explanations, but the X-axis is no longer time. Instead it represents the best-to-worst ordering of scores for each square at a single decision. In this explanation, each decision point generates a single data series, at first shown in the agent's color. Each data series contains the scores of every square on the board—even illegal ones—meaning each contains 36 rectangles. Since *BtoW* shows the scores in best-to-worst order, the leftmost rectangle corresponds to the square the agent felt to be best at the decision that just occurred, and the rightmost rectangle the worst. Hovering over any gameboard square highlights the scores associated with that square in every data series associated at all previous decision points. However, only the rectangles that are the same color as the agent are interactive—hovering them causes a gameboard square to be highlighted. Due to the sorting, if a user wants to find a specific decision, they may need to hover over some score and/or squares. Moving the game forward one step causes a new colored series to appear, the colored series from the last decision turn dark gray, and older scores become lighter.

4 METHODOLOGY

To investigate how participants would go about the Ranking Task with the explanations we have just presented, we conducted an in-person think-aloud study with 10 participants. Our RQs were:

RQ1: How did participants do on the Ranking Task?
RQ2: How many explanations and which ones did participants use when foraging for information in our interface?
RQ3: How did participants select which agents to assess while ranking?
RQ4: How did participants invest their time while ranking?

Agent Name	Noise SD	Targeted Layer	Tournament Results [W, L, D]	Win% vs RandomAgent
#1Agent	-	-	[4470, 530, 0]	98.2%
#2Agent	0.1	2	[3811, 1189, 0]	94.2%
#3Agent	0.1	3	[2712, 2288, 0]	92.9%
#4Agent	1	5	[2017, 2982, 1]	73.0%
#5Agent	1	4	[1474, 3514, 12]	81.4%
#6Agent	1	1	[503, 4484, 13]	48.0%

(a) Overall summary, aggregated tournament results, and Win% versus an agent selecting squares randomly (1000 games). Agents are named by Win% rank; for example #1Agent had the highest Win%. Tournament results are [Wins, Losses, Draws] from the perspective of the agent listed in the row.

	#2Agent	*#3Agent*	*#4Agent*	*#5Agent*	*#6Agent*
#1Agent	[726, 274, 0]	[837, 163, 0]	[967, 33, 0]	[965, 35, 0]	[975, 25, 0]
#2Agent		[668, 332, 0]	[940, 60, 0]	[965, 35, 0]	[964, 36, 0]
#3Agent			[526, 474, 0]	[913, 87, 0]	[778, 222, 0]
#4Agent				[592, 408, 0]	[858, 141, 1]

(b) Upper diagonal of matchup matrix, showing results from Table 1a broken down per pair of agents.

Table 1: Ground truth, results from large round-robin tournament.

After receiving IRB approval, we recruited participants by posting flyers around the community. All our participants gender identified[3] as either woman (6) or man (4), and had ages ranging from 20-68. They had a variety of academic backgrounds: two Art, two CS, one English, one Finance, and four different kinds of engineering. Four were associated with a branch of the military.

4.1 The Domain

Our domain was MNK games, primarily because of the strong empirical controls it afforded. MNK games are a generalization of Tic-Tac-Toe (3-3-3), with which most people are familiar. In MNK games, each player alternates placing their piece (X or O) in an attempt to put their pieces in a sequence of length K on a board of size $M \times N$. In our study, we used 9-4-4, in which a player tries for a sequence of length 4 on a 9×4 board.

Because MNK games have simple and known transition models, we programmed a strong simulator which encodes in 2 bits the 3 states of each square—opponent controlled, friendly controlled, empty. Further, the position tree has a bounded depth because eventually the board will fill. This allowed us to estimate the quality of non-terminal states using random rollouts, a property AlphaGo utilized [85].

Other researchers have used MNK (e.g., [1]), partially because *"people's intuitive priors (three-in-a-row is good) happen to be correct"* [95]. While the Go domain has a similar representation, MNK rules are much simpler, making it well-suited for HCI studies.

4.2 Manipulating agent "quality": Mutant Agent Generation

In Section 1, we pointed to the need to systematically control our manipulation—here, agent quality—which we accomplished through mutating the agent in controlled ways. First, we trained an agent to serve as the base agent. To do so, we pitted the CNN agent against a random one, used an Adam optimizer learning rate at .0001, regularized at .00001, and played games. After 125,000 games the agent was able to defeat a random agent 98.2% of the time (Table 1a). While nowhere near optimal, this level of performance was sufficient for our study because the network provided outputs accurate enough for sensible explanations.

Next, we used the base agent to generate mutant agents in the following way:

(1) Copy the neural network found in the base agent.
(2) Pick a layer in the neural network.
(3) To the network weights found on that layer, add Gaussian noise with $mean = 0$ and varying SD (we used [.01, .1, 1, 10]).
(4) Save the noisified weights.

Applying this process to our six-layer network with four noise parameter values created 24 agents, from which we chose five that spread evenly to join the base agent in the pool participants observed (as shown in Table 1a; each step down the ranking equals ≈700 fewer wins).

4.3 Procedure

We conducted a think-aloud study, one participant at a time, in our lab. Participants' task was to rank 6 agents according to which they *"think is the 'best' agent to the one that's the 'worst'."* To obtain ground truth, we used a large round-robin tournament in which the agents played against each other (Table 1b), as in Kim et al. [43]. Participants did not know the ground truth. We randomized assignment of "jersey colors", which also served as the agent's "public" name: Orange, Pink, Green, Vermilion, Sky, and Blue (Figure 3 shows accessible colors from [102]).

Before the main task, we gave participants a tutorial on the game, agents, and explanation, then conducted pre-task questionnaires collecting participant information and the first two AAR/AI steps, which define rules and agent objectives.

[3] We asked, *"What gender (if any) do you identify yourself by (check all that apply)?"* and offered the following options: Man, Woman, Non-Binary, Self-Report, and Prefer not to state [80]

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
#1Agent Loss						↓	↓			↓↓↓
#2Agent Loss			↓		↓↓	↑	↑			↑
#3Agent Loss	↓	↓↓	↑		↑	↓	↓			↑
#4Agent Loss	↓↓		↓		↓	↑	↑	↓↓		↑
#5Agent Loss	↑↑	↑↑	↑		↑↑			↑		↓
#6Agent Loss	↑							↑		↑
Total Ranking Loss	6	4	4	0	4	4	4	4	0	8
Pigeonhole Score	2	4	2	6	2	2	2	3	6	0

Table 2: Each participant's losses per agent, with agents ordered by their true rank in the first column. The arrows (↑, ↓) indicate how much worse or better participants thought each agent was than their true rank. Losses of only 1 (highlighted in light gray) show where a participant was off by only one rank. Dark gray cells highlight where a participant's ranking of that agent differed from the agent's true rank by more than one. As the table's prevalence of light colors show, overall the participants were not far off in their rankings.

During the main task, the participant stepped a game through its decisions to its conclusion while thinking aloud about the two agents' performances. The researcher then provided the AAR/AI questions on paper, but posed at the granularity of entire games instead of individual decisions as per prior work [20, 55]. The AAR/AI questions asked (1) what happened in the last game; (2) what good/bad/interesting things they observed; (3) whether/how the explanation helped them understand why that AI did the things it did; and (4) changes they recommend in the AI's decisions. They then rated both agents.

These forms were a valuable data collection artifact for us, but also served as memos participants wrote to themselves. If participants expressed confusion about what to write, the researcher mentioned that they would be retaining that form for reference, and encouraged them to write anything they might want to remember later.

After completing a form, the researcher asked if the participant was, *"...ready to do a preliminary (re)ranking, OR if they wanted to see more games—and if so, what configuration?"*. This portion of the study delivers AAR/AI steps 4 and 5 because the contents of our form cover[4] What and Why. Filling out the form is itself a learning formalization step, delivering Step 6.

Whenever the participant was ready to submit a final answer, they stopped the timers; then we conducted a short interview about their experience. The creation of the ranking delivers the formalization described in AAR/AI Step 7, intended to cover *all* observations in the session.

At the conclusion of the study, we compensated participants $20 USD, then asked if they wanted to know the "right answer." Everyone did, so we showed them the data found in Table 1a and revealed the randomized mapping between public and private names.

All of our study materials, including the scripted procedure and the source code for the system they used, are available in the Supplemental Documents accompanying this paper. Among these details is a list of slight changes we made to the interface during data collection (e.g., implementing the rewind slider, changing colors, bug fixes, etc).

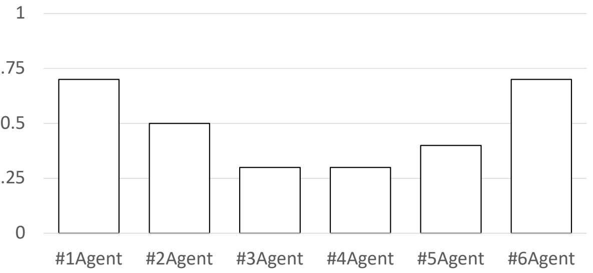


Figure 6: The fraction of participants (y-axis) who correctly ranked each agent. The U shape points out how much more successful participants were with the top/bottom agents than with the middle agents.

5 RESULTS RQ1: HOW WELL DID PARTICIPANTS RANK THE AGENTS?

Two participants ranked the agents perfectly, and several others also had a fair degree of success on the Ranking Task. We measured their success using two metrics.

The first metric, the Margin Ranking Loss[5], measures how close participants came to a perfect ranking. The Margin Ranking Loss computes for each agent $|rank_p - rank_t|$, where $rank_p$ is the rank that the participant assigned the agent, and $rank_t$ is its true rank (Table 1a).

Table 2 depicts each agent's losses with the number of arrows (↓, ↑) in each cell. The direction indicates when participants ranked the agent too high (↑) or low (↓), and the sum of the number of arrows per column shows each participant's total loss. For example, P01 incurred a loss of 1 for #3Agent by ranking it 4^{th}. Since there are 6 arrows in P01's column, their total loss was 6. Of participants' 31 losses, 23 were "off-by-one" errors (a single ↑ or ↓), half of which were adjacent agent rankings swapped, such as P06 and P07 swapping #1Agent with #2Agent.

The second metric of participant success was the number of agents they placed into the correct rank (maximum: 6). We call

[4] We chose to omit Step 3 in an effort to streamline the process and because it was not very meaningful at the game/agent level (i.e. What was supposed to happen? It was supposed to win).

[5] https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html

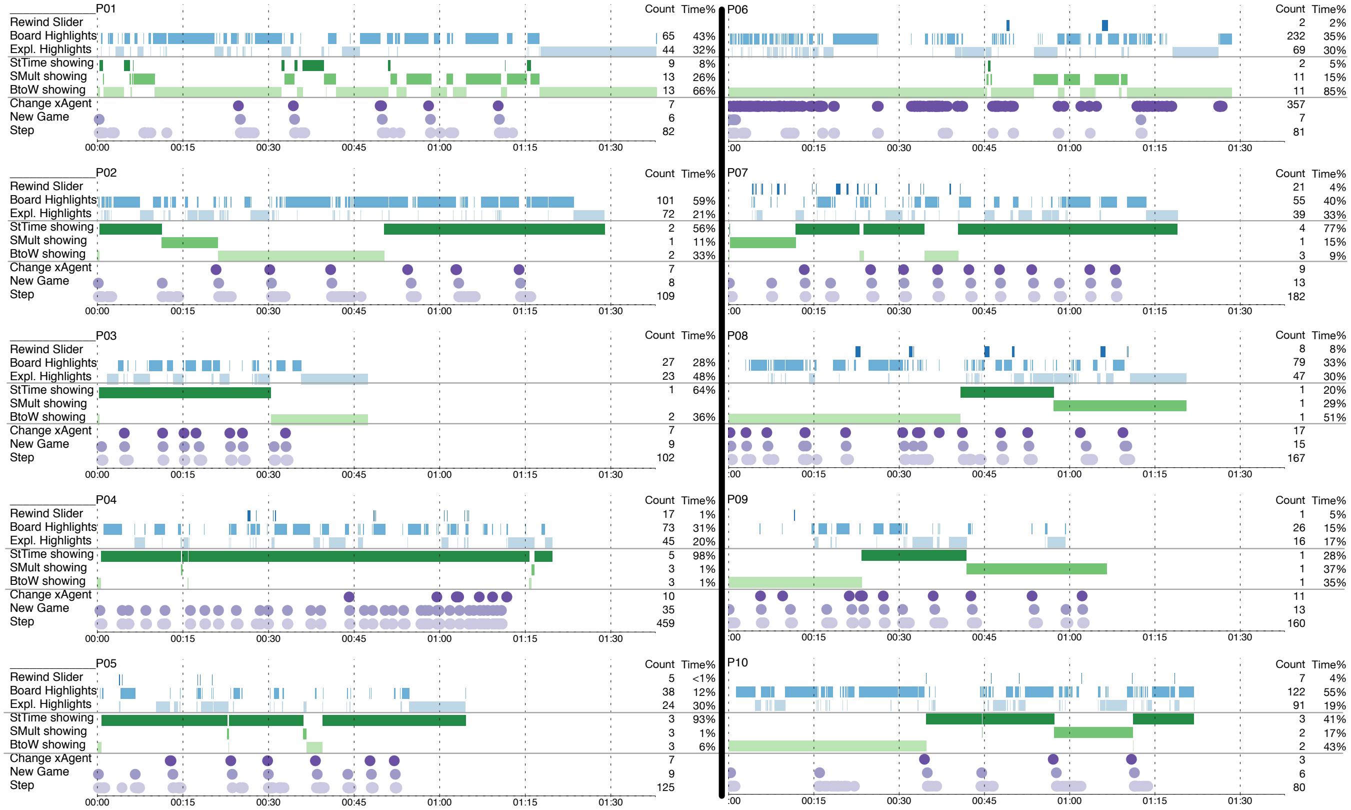


Figure 7: Timelines of each participant's events, with minutes into the main task on the X-axis. The top 3 rows (blue) show participants' interactions with the explanation. The middle 3 rows (green) show which explanation is currently visible. The bottom 3 rows (purple) show participants' interactions with game state (e.g., changing which agents are playing, which game, or advancing through game states). The text summaries show the number of instances of each event (Count) and the percentage of the participant's total task time spent in that event (Time%).

this the "pigeonhole score," per the mathematical pigeonhole principle [30]. Each participant's pigeonhole score is the number of empty cells per column in Table 2.

While Table 2 emphasizes pigeonhole success *by participant*, Figure 6 emphasizes participant pigeonhole success *by agent*. As the figure shows, the top and bottom agents were easiest for participants, with 7 participants ranking them correctly. The most difficult were #3Agent and #4Agent, with only 3 participants ranking one or both correctly. This illustrates the importance of considering explanation resolution Section 1—participants might not need fine explanation resolution to differentiate the top agent from the bottom, but may need high-resolution explanations to differentiate agents like #3Agent and #4Agent.

> P09: *"I'm pretty confident with [which agent] is at the top and...bottom, but these middle guys are a little fussier."*

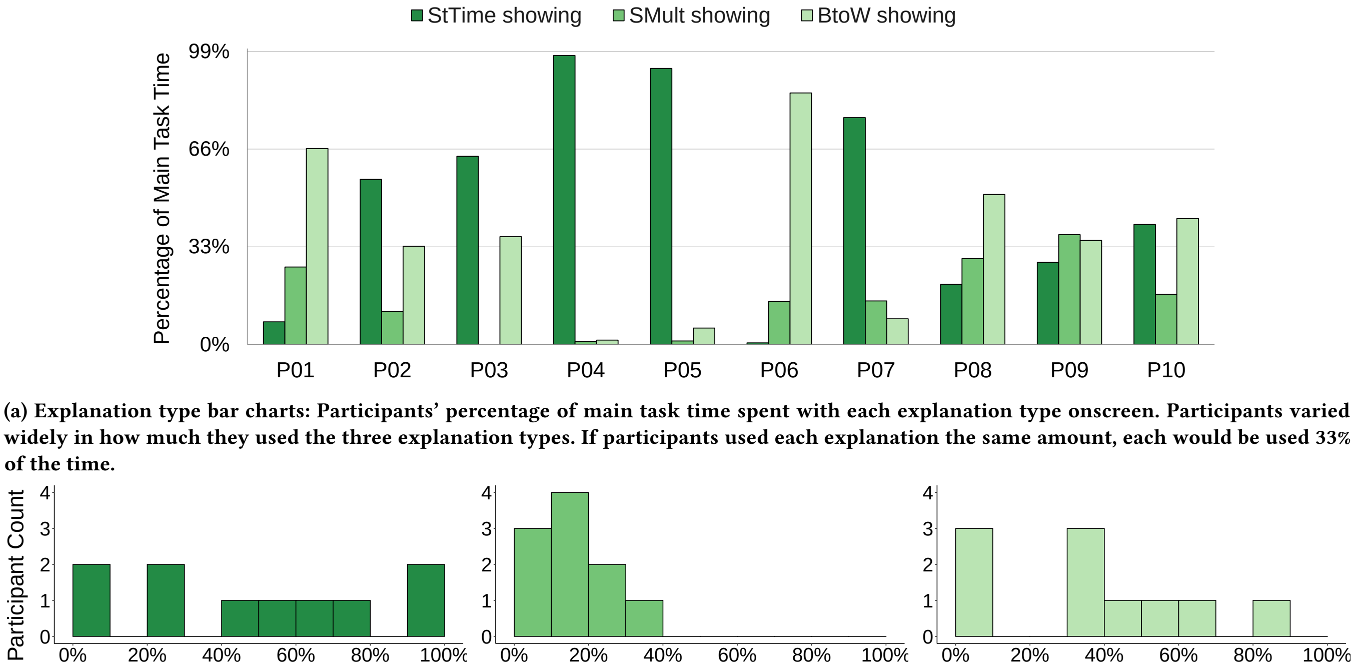
6 RESULTS RQ2: WHICH EXPLANATION TYPE(S)?

We had expected participants to try out all three explanation types. If, over time, a participant still had not tried an explanation type, the researcher would request they do so between games to encourage exposure to each one. However, as Figure 7 shows, not every participant spent much time with every explanation. For example, P03 refused to use *OnBoard* because they had decided during the tutorial that *OnBoard* was too busy. P04 remained steadfastly with *StTime*, explaining that *StTime* felt familiar due to similarities to visualizations found in sports. As Figure 7 shows, P04 and P05 gave only token glances at explanations other than *StTime*.

6.1 Participants' Explanation Diets

We can view participants' explanation choices through Information Foraging Theory's [70] concept of diets—the selection of information types that an information forager chooses to consume. Foragers' information goals determine their "ideal" diets, but what they actually consume depends on what is available in the environment. We noticed three such dietary patterns.

P04's and P05's explanation diets steadily consisted of only one explanation type throughout the task. Piorkowski et al.'s work on information foraging diets termed this the *Repeat* dietary pattern [69]. Figure 8 summarizes participants' usage patterns that were detailed in Figure 7, and both starkly reveal the Repeat diet pattern for P04



(a) Explanation type bar charts: Participants' percentage of main task time spent with each explanation type onscreen. Participants varied widely in how much they used the three explanation types. If participants used each explanation the same amount, each would be used 33% of the time.

(b) Explanation type histograms: How many participants (y-axis) used *StTime* (left), *OnBoard* (middle), and *BtoW* (right), each percentage of their time. For example, the left graph's left bar shows that 2 participants used *StTime* 0-10% of the time (disliked it), whereas its right bar shows that 2 participants used it 90-100% of the time (loved it).

Figure 8: Charts of participant usage behaviors for each explanation type.

and P05. One interpretation is that these participants stayed with *StTime* explanation because it kept giving them value. For example, after using *StTime* for about 5 minutes:

> P04: *"[#1Agent]'s gauge of win probability is flawed. It could guarantee a win earlier."*

P04 remained with *StTime* for more than an hour after that.

We term a second diet pattern, apparent in both Figure 7 and Figure 8, the *Serial Repeat* dietary pattern. In this pattern, a participant would remain with the same explanation type for a fairly long period of time (at least 10 minutes) before switching to another explanation type, where they would remain for a long period of time before switching again. The Serial Repeat dietary pattern was very common; half of the participants followed it: P02, P03, P08, P09, and P10. The pattern is visually apparent for each of these participants in Figure 7. Figure 8a shows that each of these five participants spent >=25% of their time in one explanation type and >=25% in another, corroborating these five participants' serial consumption of different explanation types.

The third dietary pattern we observed is reminiscent of Piorkowski et al.'s *Oscillate* pattern. In this pattern, a participant would start with one explanation type, then rapidly consult another explanation type to understand the phenomenon from a new perspective, then return to their first explanation type, and so on, in a series of rapid switches back and forth. Participants P01 and P06 followed this pattern often (Figure 7). As P01 explained:

> P01: *"OnBoard revealed consistent mis-scoring of obvious defensive moves. BtoW: at first, yellow seemed like it was thinking correctly about its offense, got appropriately pessimistic when missed defensive moves."*

6.2 Which explanation types?

Figure 8b shows participants' total usage for each explanation type. As that figure shows, no single explanation type outshone the others; rather, participants' preferences varied widely.

Participants also exhibited varying *degrees* of preference. For example, participants P09 and P10, both of whom followed the Serial Repeat dietary pattern, exhibited only weak preferences between the type they used the most vs. the type they used second-most. In contrast, P04 (Repeat pattern), P05 (Repeat pattern), and P06 (Repeat and Oscillate patterns) exhibited very strong preferences, each focusing almost entirely on a single explanation—but differing on *which* explanation that was.

Some participants particularly liked *StTime* and *BtoW*, with 3-5 participants using each of these two types the majority of the time, so we discuss those two explanation types first.

6.2.1 Scores Through-Time. The strengths participants saw in *StTime* were clarity, the explanation's progression over time, and ease of finding information. The main weakness they called out related to its handling of the many overlapping datapoints, which we had attempted to handle using alpha blending.

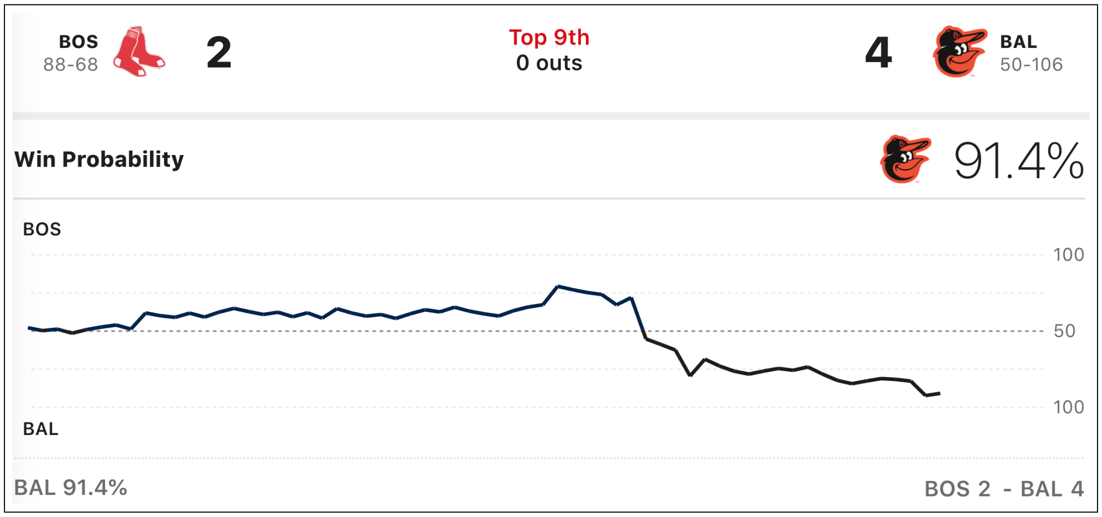


Figure 9: This ESPN chart shows analysts' estimates of two baseball teams' win probability over time, sampled near the end of the game (Source: https://www.espn.com/mlb/game/_/gameId/401229397). P04 said the *StTime* explanation felt familiar, due to similarity with sports charts like this.

> P09: *"...these columns are more clearly a separate step, so I know this was 'the third move that [#2Agent] made'. "*
> P05: *"The StTime module helps analyze steps & chronological order."*
> P10: *"And since the columns went with each turn horizontally, it was a little easier to follow as the game progressed."*
> P09: *"[StTime] still has that grey shading, which gets a little weird."*

P04 called out an advantage of the *StTime* explanation we had not expected: it reminded them of sports visualizations. For example, Figure 9 shows an ESPN visualization with the same data types on the axes, quantifies the range in the same way, and is updated with each in-game event. The main difference is that *StTime* (Figure 3) attempts to show the win probability for *all* actions, as opposed to just the one that occurred.

6.2.2 Scores Best-to-Worst. Three participants heavily used *BtoW*. Participants' remarks suggest that it may have been particularly useful in making comparisons—both among decisions and agents—but some found it confusing.

> P02: *"This BtoW move explanation helped in comparing the possible moves as they are on the same line for a particular decision."*
> P06: *"Its <current agent's> graph [BtoW] similar to [#4Agent]."*
> P09: *"I just find it confusing to read. "*

One advantage that P01 and P07 observed was *BtoW*'s ability to reveal agents' "pessimistic" expectations.

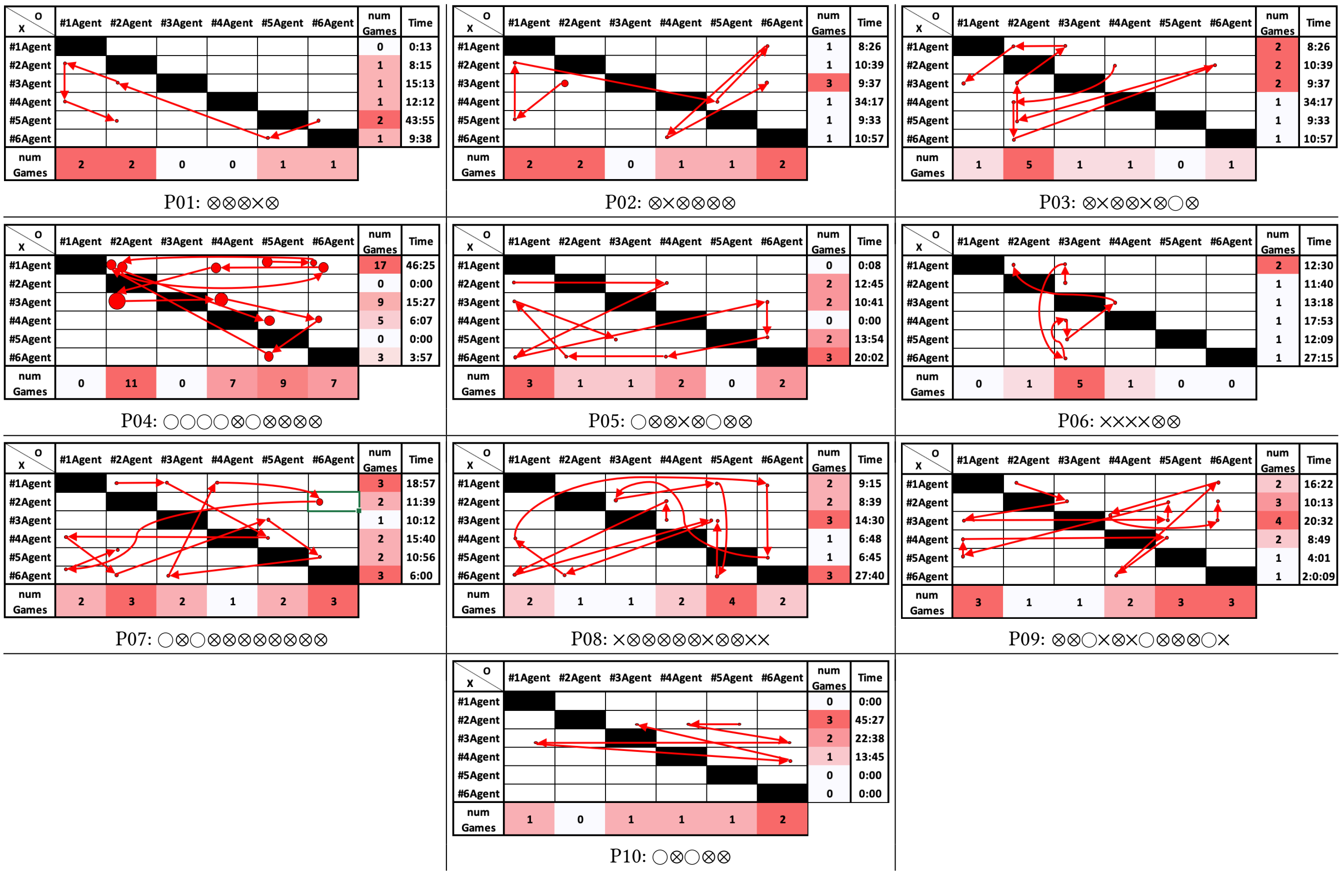
> P01: *"[#5Agent] eventually took advantage of opportunities it built over time. It made a defensive move along the way. [#6Agent]'s BtoW view revealed utter pessimism very low."*
> P01: *"BtoW: at first, [#3Agent] seemed like it was thinking correctly about its offense, got appropriately pessimistic when missed defensive moves."*
> P07: *"[#3Agent] has losing on all the turns but had multiple points where they could've had good chances."*

6.2.3 Scores On-the-Board. Explanation type *OnBoard* was no participant's clear favorite as per usage time or counts, but it seemed to play a key supporting role for some participants. Participants P01, P06, P07, and P08 all used *OnBoard* as their second-choice explanation, using it 10–30% of the time. In particular, P01 and P06, who both used the *BtoW* explanation the most (66% and 85%, respectively), brought up the *OnBoard* explanation as often as they brought up *BtoW* (13 and 11 instances each, respectively, in Figure 7).

> P06: *"The OnBoard/BtoW were similar to [#6Agent] (they both lost)."*
>
> P06: *"Started using OnBoard → lost a little confidence in [#4Agent] when looking at OnBoard."*
>
> P01: *"[#1Agent] seemed to have better diagonal defense than horizontal as per OnBoard."*
>
> P01: *"OnBoard revealed consistent mis-scoring of obvious defensive moves [for #3Agent]."*

One advantage participants particularly cited for *OnBoard* seemed to stem from the graph being "clean"—the colors map consistently to the agent colors and *OnBoard* is the only explanation in our group that is free of overlap. But others pointed to the difficulty of knowing where to look at any particular time.

> P09: *"There wasn't as much visual noise, like with the other things where there were different shades of grey indicating how old things were, it was just here's a little dot, and this represents a move. It just seemed cleaner I guess."*
>
> P08: *"Visually you could see how each one was doing."*
>
> P10: *" ...its [OnBoard] just less easily decipherable in a quick glance."*
>
> P01: *"Some of the patterns in the OnBoard view were standing out to me as potentially meaningful, but not in a way I could capitalize."*



O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							0	0:13
#2Agent							1	8:15
#3Agent							1	15:13
#4Agent							1	12:12
#5Agent							2	43:55
#6Agent							1	9:38
num Games	2	2	0	0	1	1		

P01: ⊗⊗⊗×⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							1	8:26
#2Agent							1	10:39
#3Agent							3	9:37
#4Agent							1	34:17
#5Agent							1	9:33
#6Agent							1	10:57
num Games	2	2	0	1	1	2		

P02: ⊗×⊗⊗⊗⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							2	8:26
#2Agent							2	10:39
#3Agent							2	9:37
#4Agent							1	34:17
#5Agent							1	9:33
#6Agent							1	10:57
num Games	1	5	1	1	0	1		

P03: ⊗×⊗⊗×⊗○⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							17	46:25
#2Agent							0	0:00
#3Agent							9	15:27
#4Agent							5	6:07
#5Agent							0	0:00
#6Agent							3	3:57
num Games	0	11	0	7	9	7		

P04: ○○○○⊗○⊗⊗⊗⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							0	0:08
#2Agent							2	12:45
#3Agent							2	10:41
#4Agent							0	0:00
#5Agent							2	13:54
#6Agent							3	20:02
num Games	3	1	1	2	0	2		

P05: ○⊗⊗×⊗○⊗⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							2	12:30
#2Agent							1	11:40
#3Agent							1	13:18
#4Agent							1	17:53
#5Agent							1	12:09
#6Agent							1	27:15
num Games	0	1	5	1	0	0		

P06: ××××⊗⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							3	18:57
#2Agent							2	11:39
#3Agent							1	10:12
#4Agent							2	15:40
#5Agent							2	10:56
#6Agent							3	6:00
num Games	2	3	2	1	2	3		

P07: ○⊗○⊗⊗⊗⊗⊗⊗⊗⊗

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							2	9:15
#2Agent							2	8:39
#3Agent							3	14:30
#4Agent							1	6:48
#5Agent							1	6:45
#6Agent							3	27:40
num Games	2	1	1	2	4	2		

P08: ×⊗⊗⊗⊗⊗×⊗⊗××

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							2	16:22
#2Agent							3	10:13
#3Agent							4	20:32
#4Agent							2	8:49
#5Agent							1	4:01
#6Agent							1	2:0:09
num Games	3	1	1	2	3	3		

P09: ⊗⊗○×⊗×○⊗⊗⊗○×

O X	#1Agent	#2Agent	#3Agent	#4Agent	#5Agent	#6Agent	num Games	Time
#1Agent							0	0:00
#2Agent							3	45:27
#3Agent							2	22:38
#4Agent							1	13:45
#5Agent							0	0:00
#6Agent							0	0:00
num Games	1	0	1	1	1	2		

P10: ○⊗○⊗⊗

Figure 10: How participants selected agents to assess throughout the main task, expressed as the red path through the matchup matrices, as expressed by the "New Game" data on Figure 7. Participants' path steps in these matrices are: × (vertical in the matrix) changes the X-agent. ○ (horizontal) changes the O-agent. ⊗ (diagonal) changes both agents. Size of the red dot reflects how long (# games) a participant stayed with the same pairing.

6.3 Implications for Interactive XAI and for XAI Empirical Methods

Our results do not suggest that any of these explanation types alone were the explanation of choice for a majority of participants. Some participants seemed to use all three types in almost equal amounts, whereas others used multiple types as complements—so no one type was able to fit all. This echoes earlier findings by Anderson et al. [5], who reported similar effects in a different domain with different explanation types (Saliency vs. Rewards vs. both vs. neither). Since our explanations and domain are both different from Anderson et al.'s, the similarity of these results suggests more generally that "one size does not fit all" may be a finding that is not specific to particular explanations or domains. This in turn suggests that interactive XAI systems may need to support users who wish to flexibly switch among multiple explanation types at will, or view multiple simultaneously.

From a research methods perspective, XAI empirical studies often are designed to compare different kinds of explanations as single treatments vs. a control of *no explanations*, to understand which of several explanations is *best*. However, our results suggest that such designs do not take into account individuals whose workflow includes using *multiple* explanations as *complementary* tools. In order to capture possible effects here, XAI researchers may benefit from a design using a full ablation of explanations. Unfortunately, fully ablating features causes the number of treatments to grow as a factorial in feature count. Latin Square experimental designs (see Section 5.3 in [47]) may be a useful strategy to reduce this empirical cost.

7 RESULTS RQ3: WHICH AGENTS TO ASSESS, AND HOW?

Ranking involves comparing one agent against another. If we view each such comparison as a test, choosing agent pairings is analogous to a test selection problem.

Figure 10 illustrates how participants selected agent pairings over time. In a pairing, the X-player is the agent explaining itself (eXplaining agent, or *X-agent*), and the O-player is their Opponent (*O-agent*). Whenever a game ended, most participants tended to favor changing both the X-agent *and* the O-agent—diagonal moves in the figure (and ⊗ beneath the chart). However, some participants,

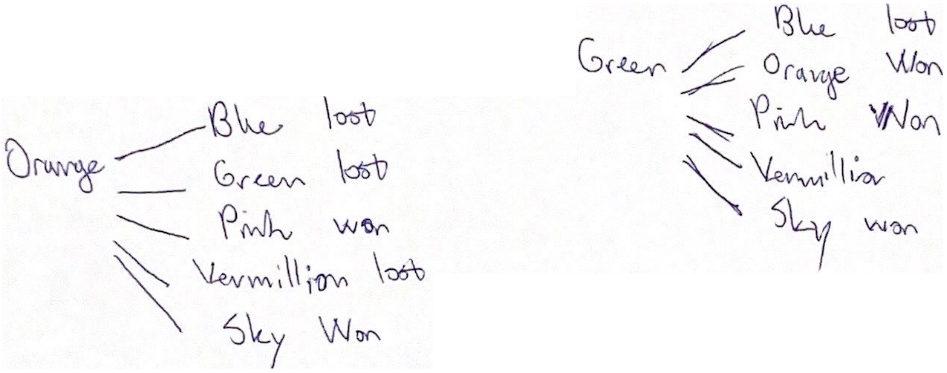


Figure 11: Two of six stumps drawn by P09 to assist in finalizing ranking. Each of these was initially created after P09 had generated a hypothesized final ranking, and written in that order. Then, using this artifact, P09 selected a few more agent pairs to assess before declaring the task complete.

such as P04 and P10, held the X-agent fixed for quite awhile, as indicated by many horizontal lines in the figure (O-agent changes, shown with ○ beneath the chart). Other participants such as P06 and P08 did the opposite, holding the O-agent fixed while changing the X-agent (indicated by many vertical lines and × beneath the chart).

7.1 Keeping Agent Pairs Synchronized

We had expected participants to take a “single-threaded” approach: start a game (“thread”), see that game through to completion, answer the AAR/AI questions, then move on to another game—in essence *terminating* the first thread. All of the participants did this except one, who instead used a “multi-threaded” approach.

P06 maintained *multiple* game threads simultaneously. P06 did so by opening each X-agent’s tab, creating a game with identical settings for each, then stepping two decisions in every game in synchrony. Since changing tabs changed both the game and the X-agent—analogous to *sleeping* the thread—the result was that P06 could inspect an explanation (usually *BtoW*, but often *OnBoard*) for the last decision from each agent, then look at the same explanation style for the last decision for the next agent, and so on. P06 continued in this way throughout the study session, allowing exactly two decisions each time and cycling through all the game threads to examine each X-agent’s explanation, synchronously across all the threads. In following these threads, P06 switched the X-agent a total of 357 times (Figure 7)—over 20 times as many as the second most frequent (P08’s 17 X-agent switches).

An advantage of P06’s approach is that it held the variable of time fixed across all games, ensuring commensurate states in terms of progress through the game. P06 also held fixed the O-agent and turn order (O-agent playing first), which had the effect of holding fixed the *difficulty* the X-agent faced. By holding as many variables as possible fixed, P06 had a more equivalent basis of comparison among the different agents than other participants did.

This means of comparison helped P06 resolve a misconception. Upon seeing each X-agent’s first explanation, P06 hypothesized to the researcher that seeing a high slope in the *BtoW* explanation indicated that the X-agent was good. However, upon seeing the explanation evolve after several decisions, P06 was able to identify that hypothesis as incorrect.

7.2 Sampling Uniformly vs Focusing on the King of the Hill

Several participants opted for a fairly uniform distribution of games assessing each agent: P01 did so mostly using *BtoW*, P03 using *StTime* and *BtoW*, and P05 mostly used *StTime*. Some of these participants ran out of time, but P05 submitted rankings early.

In contrast, some participants used a process reminiscent of Selection Sort—look for what might be the best X-agent, verify its “bestness” against several O-agents, eliminate it from consideration, and repeat. Figure 10 evidences this for P04’s perusal of the *StTime* explanations via their many horizontal moves in the top row. P10 used all three explanation types to follow the same process as P04:

> P10: *“At this point, in my opinion it is pretty clear cut that Orange and Blue are the smartest... This is how I would start at least: I would pit all of them against Blue, and switch which one goes first, and do at least one game each way like that, just to see what the other agents do against what I consider to be the smartest agent.”*

7.3 “Build-your-own” visuals

Two participants built their own visuals to track their progress beyond what they wrote on the AAR/AI forms along the way. For example, having solidified a ranking from initial observations involving all three explanation types, P09 drew a forest of six stumps, each with a different agent at the root (Figure 11). The stump appeared sideways, so the children were ordered by the hypothesized ranking, each containing a possible O-agent, with match results next to it if available. With this visual arrangement, P09 quickly determined blind spots and evaluated which wins were hard fought. With this method, P09 achieved a perfect ranking.

P08 kept a similar list of which agent won games—but omitted recording the losers. Later this omission seemed to cause confusion, as P08 conflated an agent that was too bad to win with the “untested” agents.

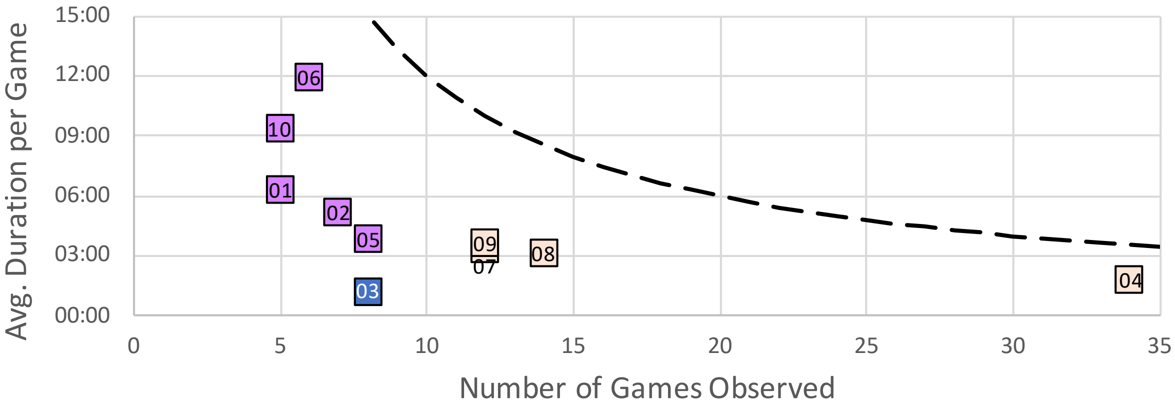


Figure 12: The number of games each participant observed (x-axis) and how long they watched each game on average (y-axis). The numbers are their PID numbers. Since they had 2 hours to complete their task, participants had to generate strategies to maximize the value of the information they received per time cost. Four participants watched as many games as possible (*ManyGames* strategy ▢), thereby having less time to spend per game (median: 03:10), but instead viewed more games (median: 13). Five participants watched each game carefully (*ThoroughGames* strategy 🟪) so had time for fewer games. One participant watched few games, but did not spend long on them (8) (*Alternate* strategy 🟦). The dashed line represents a theoretical maximum within 2 hours.

> P08: *"I want to know a little more about [#6Agent]... [#6Agent] has never won anything*
> <Researcher: Is that because it is bad or because you haven't watched it?>
> *I think I haven't watched it "*

However, P08 was wrong, #6Agent had been in 3 of the 7 games P08 had observed.

7.4 Implications for Interactive AI

Achieving the synchronization of agent-pairings that P06 sought was straightforward: P06 simply controlled the order in which they used the different interface affordances. However, our implementation of the AAR/AI component was not perfectly suited to this approach. We triggered the AAR/AI questions whenever participants finished a *game* (Section 4), but for P06's multi-threading strategy, AAR/AI's Step 6 "Formalize Learning" would have been more appropriate after every cycle of comparing all the agents for a *decision*. By mandating the formalization of learning occurring after each game instead of each workflow cycle as it naturally occurred, it is likely we disrupted P06's process. An open question for designers of interactive XAI+AAR/AI systems is: how to devise ways to trigger AAR/AI's learning formalization steps in ways appropriate to the current user's strategy?

In Section 5 we showed that it is not equally difficult to rank each item in the Ranking Task. Uniform sampling approaches aimed more at each ranking being equally difficult—thus needing equal attention. Users like these may benefit from features that guide them toward discovering, tracking, and quantifying agents performing very similarly, to help direct their attention to these more difficult portions of the Ranking Task. Similarly, King of the Hill approaches might benefit from such affordances by finding the best agent faster.

The fact that two participants built their own visuals suggests a need to give users a way to track their progress. One possibility would be to include a matchup matrix in the interface similar to Figure 10, supplemented by optional annotation/commenting capabilities. Then, for example, an updated matrix could appear after each game summarizing results observed thus far and clicking a cell could be an alternative interaction to select a pair of agents to assess.

8 RESULTS RQ4: HOW DID PARTICIPANTS INVEST THEIR TIME WHILE RANKING?

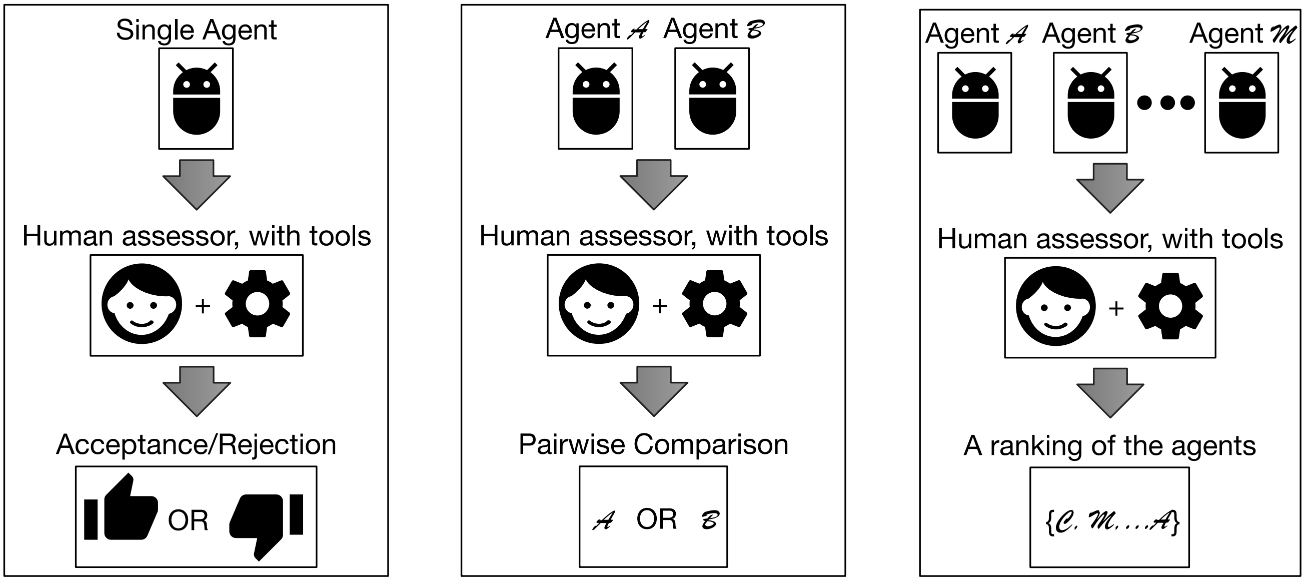
Filling the matchup matrix from Section 7 would have taken $O(n^2)$ games—but participants only had 2 hours for the task. The time limit added constraints on choices for how many games participants saw from each pairing, and how long to spend per game.

8.1 Invest in Many Games

ManyGames participants spent less time per game ($\leq$ *Median*(*Avg. Duration*)) so as to see more games ($>$ *Median*(*# Games*)) (Figure 12) . Four participants invested this way—including the two who ranked perfectly (P04 & P09). On average, *ManyGames* investors had an average pigeonhole score of 4.25, and they incurred lower losses (avg. = 2) than their peers.

ManyGames investors also seemed to gain a robustness against "underdog" victories[6] warping their rankings. To illustrate, P04 observed 34 games, 24% of which were underdog victories, including the best agent (#1Agent) *losing* to the worst (#6Agent)! Despite this, P04 ranked all agents perfectly, possibly because they observed

[6] In Table 1b, an underdog victory is where a lower-ranked agent wins against a higher-ranked opponent (i.e., #3Agent defeating #1Agent).



(a) Acceptance testing: Provided *one* input item, assessors determine fitness *"for the purpose."* [27].

(b) Comparison testing: Provided *two* input items (as in [38]), assessors determine "which is better."

(c) Ranking (our proposal): Provided *many* input items, assessors fully order them.

Figure 13: Three notional views of measuring the quality of explanation systems. Note that each takes as input an agent and situation (e.g. the agent has a wall adjacent), allowing the human to rank/accept with respect to a different property (e.g. speed or win count).

#1Agent defeat #6Agent in 4 additional games. P09 also ranked the agents perfectly, and when asked about what they would do if given more time, their response was to repeat observations:

> P09: *"I might replay a few of them that I have already played, just to see if I get the same results."*

8.2 Invest Thoroughly in Games

Five participants spent more time per game[7] (> *Median*(*Avg. Duration*)) but in turn saw fewer games (≤ *Median*(*# Games*)). *ThoroughGames* participants had an average pigeonhole score of 2 and incurred higher losses (avg. 5.6) than their peers.

The *ThoroughGames* investors also seemed susceptible to underdog victories, and at least one seemed aware of it:

> P10: *"[#1Agent] might have been smarter, I've only seen it in one game... I kinda wish I could have seen it in one more."*

Here, P10 ranked the *best* agent as third, having only observed it lose to #3Agent, with #3Agent as the explaning agent.

Still, an advantage of the *ThoroughGames* approach is that these participants reflected more deeply on the explanations:

> P02: *"At 11th move, the Orange agent have not selected the best move which would result in winning for the agent."*
> P01: *"Pink had very low scores for obvious defensive moves that it missed."*
> P10: *"Blue seemed to rank all moves properly, except the last [winning move] which it still didn't rank as 100%"*

8.3 Implications for XAI research

Participants' trade-offs between maximizing the number of games observed (*ManyGames*) or the time spent per game (*ThoroughGames*) were reminiscent of Rader et al.'s [72] methods of improving transparency in an intelligent system: 1) repeated experiences with a system and 2) explanations into the system's thinking. This raises a potential conflict with the XAI researchers' goals: How do we collect good data about our explanations from participants who just want to use the system and ignore the explanation?

To illustrate, while we previously highlighted how forthcoming *ThoroughGames* investors seemed as research participants, we had some difficulty obtaining high quality written data from *ManyGames* investors. Concretely, only 1 of 4 *ManyGames* investors had a written response we coded as Explanation Interpretation. Further, *ManyGames* investors often declined or replied "Nothing" when asked what information present in the explanation helped them (3 times on average)—despite evidently hovering and seeming to look at it. This suggests XAI researchers may want to rely on data sources which are not self-reported to improve data quality from people like our *ManyGames* investors, e.g., using direct and indirect measures like eye tracking or the ranking task.

9 DISCUSSION

9.1 What Good Is the Ranking Task?

We devised the Ranking Task to fill gaps in existing catalogs of empirical XAI tasks, e.g., those surveyed by Hoffman et al. [32].

[7] Their games did not take more steps; there was no significant difference in steps/game across the participants (ANOVA, F(9,101)=0.297, p=.974).

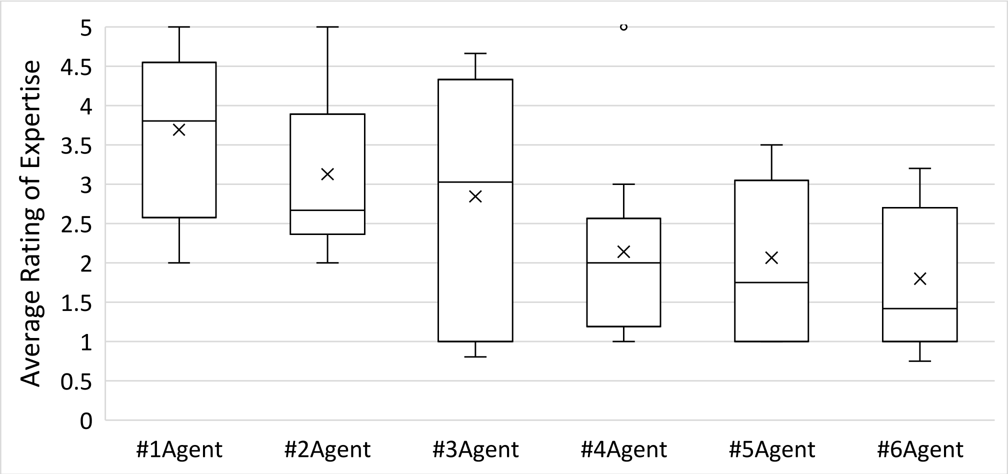


Figure 14: Distributions of participants' average ratings (y-axis) of the 6 agents' expertise (x-axis). Notice that in Figure 6, participants' accuracy ranking the agents was U-shaped, but participants' in-situ ratings of the agents expertise here was more aligned with the agent quality (i.e., participants rated 6-MNL worst, and #1Agent best).

For example, Anderson et al. [5] reported situations in which asking participants to *predict* an agent's next decision produced very noisy results. High variability in feature/action space may be one possible noise source in the prediction task. Once this space is big enough, the probability that a participant selects the correct action becomes vanishingly small [19]. Prediction tasks can benefit from "partial credit" for when the participant's chosen action is "close", although defining action similarity remains challenging. The Ranking Task avoids this issue by acting at higher granularity than individual actions.

Figure 13 illustrates measuring advantages the Ranking Task brings to XAI researchers. Acceptance testing (left) [27] is challenging to ground truth, as it can be difficult to define criteria for the assessor's acceptance. Comparison testing (middle) [38] resolves most of the problems obtaining ground truth, but remains a low resolution measure (1-bit). Ranking (right) can be ground-truthed in a manner similar to comparison testing, and provides a higher resolution measurement of an assessor's ability to differentiate agents.

Random guessing at Comparison testing will be 50% correct, which makes scientific inference hard without large sample sizes. Meanwhile, applying `MarginRankingLoss` to participant rankings "puts more marks on the ruler" in terms of allowing more precise measurement. The output from this loss function is 0 for the perfect ranking, ranging up[8] to a function $\in \Theta(n^2)$ for n agents. Thus, our loss describes a direct measure of participants' performance at the task, as opposed to relying on self-reported data.

However, perhaps there are other ways to generate a ranking, rather than requesting one explicitly. For example, at the end of each game, participants were asked to rank the expertise of both agents in the game on a scale of 1 (novice player) to 5 (expert player). After averaging participants' expertise scores for each of the 6 agents, Figure 14 illuminates another possibility: inducing a ranking based on participants' in-situ ratings along the way. In particular, the averages of participants' along-the-way expertise scores (Likert 1–5) were more reflective of the true ranking of the agents, which raises the question of the best way to solicit the ranking: at task end, incrementally along the way, or some combination?

9.1.1 *Case Study: Calculating Explanation Resolution.*

Our study was *not* designed for comparative statistics, but to demonstrate calculation of explanation resolution, we proceed in this case study as though it was. A comparative study would have assigned explanation-type treatments; here we approximate this by binning participants into an explanation-type treatment if they used that type at least half as often as that participant's most-used. Table 3 shows the results of binning this way and the average losses across participants associated with each treatment. We interpret this value as a direct measure of explanation resolution. With this interpretation, we would conclude that *OnBoard+BtoW* had the worst (lowest) explanation resolution, and that *StTime* had the highest. (N.B., this computation is *strictly* for demonstration purposes.)

9.2 The Ranking Task as an instance of "The Coaches' Problem"?

We find it useful to consider human analogs to problems found in evaluating AI, so we propose *The Coaches' Problem—"Given a set of players on the roster, how do we pick which ones should start?"*. When humans approach this problem with human players, they are often potential-oriented, *not* results-oriented ("*Will this player help us win games in the future?*" vs "*Did this player help us win games in the past?*"). This means that coaches evaluate beyond the stat sheets—e.g., mechanics, attitude, or injury risk.

Consider that in the Coaches' Problem, humans must often pick *before* big data is available. Thus, they observe settings such as drills and scrimmages—smaller data than the full season. And so, coaches find themselves predicting big data from small, much as AI assessors

[8] The empty ranking for n agents has loss $\frac{n(n+1)}{2}$, though the worst loss we could find using responses including *all* agents *exactly once* was $\lfloor \frac{1}{2} n^2 \rfloor$, for the backwards ranking.

"Treatment"	Participants	Average Loss, (AKA "Explanation Resolution")
StTime	[P03, P04, P05, P07]	3.5
OnBoard	[ ]	
BtoW	[P06]	4
StTime+OnBoard	[ ]	
StTime+BtoW	[P02]	4
OnBoard+BtoW	[P01]	6
StTime+OnBoard+BtoW	[P08, P09, P10]	4

Table 3: Case Study calculation of explanation resolution for an ablation of our three explanations. We used the loss instead of the pigeonhole score because of the relationship to the microscopy resolution definition discriminating neighboring points.

must. Just as better drills will allow coaches greater insight with only limited observations, so too will better explanations improve the insight of AI assessors.

Further, both coaching and AI are often *organizational* efforts conducted within *limited time* constraints, elevating the importance of boundary objects. For example, artifacts to support collaborative work (boundary objects) might help a single manager digest information from many scouts. Additionally, boundary objects might also assist scouts recalling past observations (collaborating with past/future self). To be concrete, we observe that artifacts proposed for AI evaluation (e.g., Model Cards [58]) bear significant resemblance to reports found in various sports[9], so perhaps these two communities can learn from each other.

9.3 The Ranking Task vs. AutoML

One might argue that automating the application of ML to real world problems, (AutoML [89]) reaches straight for the large scale past-facing evaluation data that we use for "ground truth" (e.g., Figure 2 in Wang et al. [99] shows ranked models). However, AutoML approaches tend to train multitudes of models, running many tests on each—sometimes daunting given the cost to train recent enormous models (e.g., [6] estimates GPT-3 cost $10M). Here, humans determining models' deployment-worthiness via explanation might be cheaper than running parallel training processes.

We view limiting the need for training as one of the most important interventions to reduce AI costs—both carbon and monetary. As such, under the mutant agent generation workflow described in this paper, measuring a new explanation does *not* require a new training process. Similarly, we chose a challenge domain with relatively low computing overhead, e.g., as opposed to Atari domains [61]. Lastly, because our tasks do not require an optimal agent, we did not need to train the agent very long, e.g. little hyperparameter tuning, short training jobs on few machines. As a result, our total compute budget was on the order of 100s of kW (running 2-3 regular desktop computers for several days).

9.4 Why Mutant Agent Generation?

Mutant Agent Generation offers a very low cost tool to create a potentially large number of agents of controllably differing quality, to support an AI testing methodology.

One source of inspiration is literature on mutation testing, first published in 1978 [16], but still used today [68]. In mutation testing, the first step is to generate mutants by manipulating the source code many times (e.g. replacing a "+" with "-"), each time creating a different mutant. Then, the quality of a testing methodology can be measured by the number of detected and "killed" mutants. Thus, we can similarly measure the efficacy of a "test suite" for AI—the person-machine team of human plus explanation—by ability to detect the presence of mutation in an agent.

Some source code mutations are harder to kill than others (e.g., replacing > with ≥ might trigger problems rarely). Similarly, adding very small amounts of noise to the network weights induces an agent encoding a policy similar to the original[10]; while large amounts of noise will produce an essentially random agent. Table 1a illustrates that the most damaged agent is on par with a random one, and that "Low" noise agents are the least damaged.

Researchers have investigated a variety of other manipulations for AI systems. As an example, instead of mutating agents, diverse agents often arise as a natural result of training, and can be used for comparison. Huang et al. [34, 35] used this strategy, finding it assisted human assessment by selecting more informative states. Other properties researchers have manipulated include opaqueness [71], complexity [71], fairness [21], and more.

Of course, there are *more* controllable manipulations available, such as choosing a specific set of neurons that seem correlated to some desired feature [73]. However, such manipulations are labor-intensive to implement because each must account for factors such as domain, task, architecture, etc. In contrast, an advantage of mutant agent generation is applicability to essentially *any* neural network[11] with little development effort[12], similar to how mutation testing can be applied in semi-automated ways [82] to essentially any source code.

10 THREATS TO VALIDITY

Every study has threats to validity [101]—in reviewing ours, we follow Yin's approach [104].

First, our findings may not generalize well. Qualitative studies like ours recruit small sample sizes to analyze individual participants in depth. As such, the strength of qualitative studies lies in revealing unforeseen, unreported phenomena. Beyond the small

[9] E.g., https://sports.yahoo.com/nfl/players/31934/situational

[10] In the limit $SD \rightarrow 0$, the Gaussian becomes the Delta function, which would result in *no change* to the weights *or* policy because the mean is 0.

[11] It may be applicable even beyond neural networks. For example, we envision analogous techniques for other types of models, such as noisifying feature weights in a linear regressor.

[12] The short function "noisifySelf" in the CNNAgent; see provided source code for an example.

sample size, other factors that preclude generalization are: focusing on a single task, domain, agent architecture, and agent pool generation strategy.

Another threat to generalizability is the MNK domain itself, which is not a common AI challenge domain. Many studies that investigate sequential decision-making agents instead use games like StarCraft (surveyed in Ontanon et al. [64]). However, StarCraft's complexity adds costs to the tutorial or participant sampling, since some experience is required. Further, episodes are long (15–60min) and difficult to experimentally control due to the player-controlled camera [67]. Lastly, although good agents exist [96], they are not publicly available. The best alternative agents are heterogeneous competition submissions (as in Kim et al. [43]), making explanation difficult. Meanwhile, the compute required to train quality StarCraft agents is infeasible for most researchers [18].

We selected MNK for several reasons. First, most people have familiarity with Tic-Tac-Toe (the 3-3-3 instance of MNK)—including all of our participants. Even without familiarity, training time is minimal because the games are simple. The shortness of the game enables the comparison needed to rank, since participants had time to watch multiple agents play. MNK also gives experimenters a high degree of control, e.g. varying task difficulty, *both* in terms of participant foraging difficulty *and* in terms of strategic complexity—by simply adjusting M, N, and K.

Still, MNK games bring the threat that they were perhaps too easy, and therefore not representative of ranking tasks that might arise in the real world. For example, MNK games could be solved by other strategies (e.g. value iteration or search). However, by studying how people assess neural networks in our toy domain, we can better prepare for more complicated problems.

One component absent from our interface is the capability to perform a "big data" analysis on the agents. Although this is an important piece of an agent assessment interface, we eliminated it from our study because (1) we established ground truth with that information, and so could not reveal it; and (2) we aim for explanation systems like ours to assess systems where "big data" cannot necessarily illuminate the best agent automatically, e.g., when large-scale deployment data is expensive to collect and/or nonexistent.

Our use of mutant agents raises another threat: mutant agent generation is not ecologically valid. Mutants might appropriately model random errors, but perhaps not *systematic* errors typical in ML applications. In future work, we could assess this threat by comparing our explanations' ability to point out differences in various agent pools, e.g. mutated agents vs. agents sampled from historical training configurations [34].

Another threat is that we asked participants to accomplish only one XAI task with only our novel explanations. Alternatives might have allowed us to compare with prior work, e.g, if participants had additionally performed tasks and/or used explanations from prior literature. However, we wanted to observe participants over time as they focused on the novel aspects of our task and explanations.

Finally, we did not control how participants went about their task, so each experienced something different. We designed the investigation without this control so as to observe their unconstrained behavior, but the lack of controls adds another threat to generalizability.

11 CONCLUSION

We investigated how 10 participants went about a new empirical task—the Ranking Task. Toward this end, we created three explanation types, scaffolded them with an adaptation of the AAR/AI process, and introduced a way to control agent variation—Mutant Agent Generation. This approach is a computationally efficient, controllable, simple, and general way to select a pool of agents that are more/less similar, by changing the amount of noise and number of agents to rank.

Our participants:

- ...ranked the agents well overall, but showed the importance of a concept we term *explanation resolution* for close differences between agents (Section 5). Fortunately, researchers can measure this quantity to reveal where an explanation type is (in)adequate (Section 9.1.1).
- ...were diverse in both the explanation types they used, and how they combined them into an information diet. The results suggest that single-explanation approaches may malnourish users who thrive on a multi-explanation diet.
- ...approached agent "test selection" (agent pairing selection) in at least four different ways: (1) synchronizing different agent pairs playing the same game, (2) sampling uniformly, (3) focusing on the "king of the hill", or (4) building their own visualizations to maintain results. Each group's success (or lack thereof) suggests the need for new affordances enabling users to track their progress through the Ranking Task.
- ...traded off number of games to observe vs. how much time to invest in each game in different ways, some favoring the former (*ManyGames*) and others favoring the latter (*ThoroughGames*). A strength *ManyGames* participants exhibited was increased resilience to underdog victory anomalies.

In addition, an important takeway for XAI researchers is that our results suggest that use of the Ranking Task can help reveal important nuances in XAI explanations' ability to support users' in their understanding of intelligent agents.

> P09: *"I ranked Orange [#3Agent] above Vermilion [#4Agent] just because as I was looking at Orange's decision making process in the graphs it made a lot of sense to me, so thats why I put Orange above Vermilion."*

ACKNOWLEDGMENTS

This material is based upon work supported by DARPA #N66001-17-2-4030 and joint support by NSF and USDA-NIFA under #2021-67021-35344.

REFERENCES

[1] Abdel-Hafiz Abdoulaye, Vinasetan Ratheil Houndji, Eugène C. Ezin, and Gael Aglin. 2018. Generic Heuristic for the mnk-games. In *African Conference on Research in Computer Science* (Stellenbosch, South Africa) *(CARI '18)*. 265–275. https://www.cari-info.org/Actes-2018/p276-286.pdf

[2] S. Amershi, M. Cakmak, W. Knox, and T. Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[3] Saleema Amershi, Max Chickering, Steven Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2015)* (proceedings of the conference on human factors in computing systems (chi 2015) ed.). ACM - Association for Computing Machinery. https://www.microsoft.com/en-us/research/publication/modeltracker-redesigning-performance-analysis-tools-for-machine-learning/

[4] 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.

[5] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett. 2019. Explaining Reinforcement Learning to Mere Mortals: An Empirical Study. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, Palo Alto, CA, USA, 1328–1334. http://dl.acm.org/citation.cfm?id=3367032.3367221

[6] Nathan Benaich and Ian Hogarth. 2020. State of AI Report. https://www.stateof.ai/

[7] Ralph Brewer, Anthony Walker, E. Ray Pursel, Eduardo Cerame, Anthony Baker, and Kristin Schaefer. 2019. Assessment of Manned-Unmanned Team Performance: Comprehensive After-Action Review Technology Development. In *2019 International Conference on Human Factors in Robots and Unmanned Systems* (Washington D.C., USA) *(AHFE '19)*. Springer Nature Switzerland AG, Cham, CHE, 119–130.

[8] Timothy A Budd, Richard J Lipton, Richard DeMillo, and Frederick Sayward. 1978. The design of a prototype mutation system for program testing. In *Managing Requirements Knowledge, International Workshop on*. IEEE Computer Society, 623–623.

[9] Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. 2019. The Effects of Example-based Explanations in a Machine Learning Interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. ACM, New York, NY, USA, 258–262. https://doi.org/10.1145/3301275.3302289

[10] Nicholas Carlini and David Wagner. 2016. Towards Evaluating the Robustness of Neural Networks. arXiv:1608.04644 [cs.CR]

[11] Nan-Chen Chen, Jina Suh, Johan Verwey, Gonzalo Ramos, Steven Drucker, and Patrice Simard. 2018. AnchorViz: Facilitating Classifier Error Discovery through Interactive Semantic Data Exploration. In *Proceedings of the 23th International Conference on Intelligent User Interfaces*. ACM, 269–280. https://www.microsoft.com/en-us/research/publication/anchorviz-facilitating-classifier-error-discovery-interactive-semantic-data-exploration/

[12] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*. 2172–2180.

[13] 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. *Cognitive Science* 13, 2 (4 1989), 145–182. https://doi.org/10.1207/s15516709cog1302_1

[14] Michael Correll, Dominik Moritz, and Jeffrey Heer. 2018. *Value-Suppressing Uncertainty Palettes*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3174216

[15] Robert Davies, Elly Vaughan, Graham Fraser, Robert Cook, Massimo Ciotti, and Jonathan E. Suk. 2019. Enhancing Reporting of After Action Reviews of Public Health Emergencies to Strengthen Preparedness: A Literature Review and Methodology Appraisal. *Disaster Medicine and Public Health Preparedness* 13, 3 (june 2019), 618–625. https://doi.org/10.1017/dmp.2018.82

[16] R. A. DeMillo, R. J. Lipton, and F. G. Sayward. 1978. Hints on Test Data Selection: Help for the Practicing Programmer. *Computer* 11, 4 (April 1978), 34–41. https://doi.org/10.1109/C-M.1978.218136

[17] Shipi Dhanorkar, Christine T. Wolf, Kun Qian, Anbang Xu, Lucian Popa, and Yunyao Li. 2021. *Who Needs to Know What, When?: Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle*. Association for Computing Machinery, New York, NY, USA, 1591–1602. https://doi.org/10.1145/3461778.3462131

[18] Jonathan Dodge. 2021. Position: Who Gets to Harness (X)AI? For Billion-Dollar Organizations Only. In *IUI Workshops*.

[19] Jonathan Dodge and Margaret Burnett. 2020. Position: We Can Measure XAI Explanations Better with "Templates". In *IUI Workshops*.

[20] Jonathan Dodge, Roli Khanna, Jed Irvine, Kin-Ho Lam, Theresa Mai, Zhengxian Lin, Nicholas Kiddle, Evan Newman, Andrew Anderson, Sai Raja, Caleb Matthews, Christopher Perdriau, Margaret Burnett, and Alan Fern. 2021. After-Action Review for AI (AAR/AI). *ACM Transactions on Interactive Intelligent Systems* (2021), 33 pages. http://web.engr.oregonstate.edu/~burnett/Reprints/TIIS21_AARAI-accepted-preprint.pdf (To Appear).

[21] Jonathan Dodge, Q. Vera Liao, Yunfeng Zhang, Rachel K. E. Bellamy, and Casey Dugan. 2019. Explaining Models: An Empirical Study of How Explanations Impact Fairness Judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. ACM, New York, NY, USA, 275–285. https://doi.org/10.1145/3301275.3302310

[22] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Xiaodong Song. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 1625–1634.

[23] Philip M Fernbach, Steven A Sloman, Robert St Louis, and Julia N Shube. 2012. Explanation fiends and foes: How mechanistic detail determines understanding and preference. *Journal of Consumer Research* 39, 5 (2012), 1115–1131.

[24] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*. 80–89. https://doi.org/10.1109/DSAA.2018.00018

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[26] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh. 2014. You Are the Only Possible Oracle: Effective Test Selection for End Users of Interactive Machine Learning Systems. *IEEE Transactions on Software Engineering* 40, 3 (March 2014), 307–323. https://doi.org/10.1109/TSE.2013.59

[27] Brian Hambling and Pauline van Goethem. 2013. *User acceptance testing: a step-by-step guide*. BCS Learning and Development, Swindon. http://cds.cern.ch/record/1619552

[28] Samer Hanoun and Saeid Nahavandi. 2018. Current and Future Methodologies of After Action Review in Simulation-based Training. In *2018 Annual IEEE International Systems Conference (SysCon)* (Vancouver, BC, CAN) *(SysCon '18)*. IEEE, New York, NY, USA, 1–6.

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

[30] IN Herstein. 1969. Topics in Algebra-Walthan.

[31] C. Hill, R. Bellamy, T. Erickson, and M. Burnett. 2016. Trials and tribulations of developers of intelligent systems: A field study. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 162–170. https://doi.org/10.1109/VLHCC.2016.7739680

[32] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for Explainable AI: Challenges and Prospects. *CoRR* abs/1812.04608 (2018). arXiv:1812.04608 http://arxiv.org/abs/1812.04608

[33] 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* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 579, 13 pages. https://doi.org/10.1145/3290605.3300809

[34] Sandy H. Huang, Kush Bhatia, Pieter Abbeel, and Anca D. Dragan. 2018. Establishing Appropriate Trust via Critical States. *IROS* (Oct 2018). https://doi.org/10.1109/IROS.2018.8593649

[35] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. 2017. Enabling Robots to Communicate Their Objectives. *CoRR* abs/1702.03465 (2017).

[36] Andrew Ishak and Elizabeth Williams. 2017. Slides in the Tray: How Fire Crews Enable Members to Borrow Experiences. *Small Group Research* 48, 3 (March 2017), 336–364. https://doi.org/10.1177/1046496417697148

[37] Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng Chau. 2018. ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models. *IEEE Transactions on Visualization and Computer Graphics* 24 (2018), 88–97.

[38] Minsuk Kahng, Dezhi Fang, and Duen Horng (Polo) Chau. 2016. Visual Exploration of Machine Learning Results Using Data Cube Analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (San Francisco, California) *(HILDA '16)*. ACM, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/2939502.2939503

[39] Minsuk Kahng, Nikhil Thorat, Duen Horng (Polo) Chau, Fernanda B. Viégas, and Martin Wattenberg. 2019. GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2019), 310–320.

[40] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1343–1352.

[41] Nathanael L Keiser and Winfred Arthur Jr. 2020. A meta-analysis of the effectiveness of the after-action review (or debrief) and factors that influence its effectiveness. *Journal of Applied Psychology* (2020).

[42] Roli Khanna, Jonathan Dodge, Andrew Anderson, Rupika Dikkala, Jed Irvine, Zeyad Shureih, Kin-ho Lam, Caleb R. Matthews, Zhengxian Lin, Minsuk Kahng, Alan Fern, and Margaret Burnett. 2021. Finding AI's Faults with AAR/AI: An Empirical Study. *ACM Transactions on Interactive Intelligent Systems* (2021). To Appear.

[43] M. Kim, K. 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.

[44] Alexandra Kirsch. 2017. Explain to whom? Putting the User in the Center of Explainable AI. In *CEx@AI*IA*.

[45] Gary Klein, Louise Rasmussen, Mei-Hua Lin, Robert R Hoffman, and Jason Case. 2014. Influencing preferences for different types of causal explanation of complex events. *Human factors* 56, 8 (2014), 1380–1400.

[46] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 5686–5697. https://doi.org/10.1145/2858036.2858529

[47] Robert Kuehl. 2000. Design of experiments : statistical principles of research design and analysis / Robert O. Kuehl. *SERBIULA (sistema Librum 2.0)* (01 2000).

[48] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 126–137.

[49] Todd Kulesza, Simone Stumpf, Margaret Burnett, Weng-Keen Wong, Yann Riche, Travis Moore, Ian Oberst, Amber Shinsel, and Kevin McIntosh. 2010. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on*. IEEE, 41–48.

[50] Todd Kulesza, Weng-Keen Wong, Simone Stumpf, Stephen Perona, Rachel White, Margaret M Burnett, Ian Oberst, and Andrew J Ko. 2009. Fixing the program my computer learned: Barriers for end users, challenges for the machine. In *Proceedings of the 14th international conference on Intelligent user interfaces*. 187–196.

[51] Xiaodan Liang, Liang Lin, Xiaohui Shen, Jiashi Feng, Shuicheng Yan, and Eric P Xing. 2017. Interpretable Structure-Evolving LSTM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1010–1019.

[52] Zhengxian Lin, Kin-Ho Lam, and Alan Fern. 2021. Contrastive Explanations for Reinforcement Learning via Embedded Self Predictions. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Ud3DSz72nYR

[53] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. 4765–4774.

[54] 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

[55] Theresa Mai, Roli Khanna, Jonathan Dodge, Jed Irvine, Kin-Ho Lam, Zhengxian Lin, Nicholas Kiddle, Evan Newman, Sai Raja, Caleb Matthews, Christopher Perdriau, Margaret Burnett, and Alan Fern. 2020. Keeping It "Organized and Logical": After-Action Review for AI (AAR/AI). In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 465–476. https://doi.org/10.1145/3377325.3377525

[56] Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the State of the Art of Evaluation in Neural Language Models. In *ICLR*. OpenReview.net.

[57] Ronald Metoyer, Simone Stumpf, Christoph Neumann, Jonathan Dodge, Jill Cao, and Aaron Schnabel. 2010. Explaining how to play real-time strategy games. *Knowledge-Based Systems* 23, 4 (2010), 295–301.

[58] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 220–229. https://doi.org/10.1145/3287560.3287596

[59] John E. Morrison and Larry L. Meliza. 1999. *Foundations of the After Action Review Process*. Technical Report. Institute for Defense Analyses. https://apps.dtic.mil/docs/citations/ADA368651

[60] W. James Murdoch and Arthur Szlam. 2017. Automatic Rule Extraction from Long Short Term Memory Networks. *ArXiv* abs/1702.02540 (2017).

[61] Johan Samir Obando-Ceron and Pablo Samuel Castro. 2021. Revisiting Rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1373–1383. http://proceedings.mlr.press/v139/ceron21a.html

[62] Matthew L Olson, Roli Khanna, Lawrence Neal, Fuxin Li, and Weng-Keen Wong. 2021. Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence* 295 (2021), 103455.

[63] Matthew L. Olson, Thuy-Vy Nguyen, Gaurav Dixit, Neale Ratzlaff, Weng-Keen Wong, and Minsuk Kahng. 2021. Contrastive Identification of Covariate Shift in Image Data. In *2021 IEEE Visualization Conference (VIS)*. IEEE.

[64] 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 Transactions on Computational Intelligence and AI in Games* 5, 4 (Dec 2013), 293–311. https://doi.org/10.1109/TCIAIG.2013.2286295

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[66] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore. *Proceedings of the 26th Symposium on Operating Systems Principles - SOSP '17* (2017). https://doi.org/10.1145/3132747.3132785

[67] 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 *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. ACM, New York, NY, USA, 225–237.

[68] Goran Petrović and Marko Ivanković. 2018. State of Mutation Testing at Google. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice* (Gothenburg, Sweden) *(ICSE-SEIP '18)*. ACM, New York, NY, USA, 163–171. https://doi.org/10.1145/3183519.3183521

[69] David J Piorkowski, Scott D Fleming, Irwin Kwan, Margaret M Burnett, Christopher Scaffidi, Rachel KE Bellamy, and Joshua Jordahl. 2013. The whats and hows of programmers' foraging diets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3063–3072.

[70] P. Pirolli. 2007. *Information Foraging Theory: Adaptive Interaction with Information*. Oxford Univ. Press.

[71] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. 2021. *Manipulating and Measuring Model Interpretability*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445315

[72] Emilee Rader, Kelley Cotter, and Janghee Cho. 2018. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[73] Ivet Rafegas, Maria Vanrell, Luís A. Alexandre, and Guillem Arias. 2020. Understanding trained CNNs by indexing neuron selectivity. *Pattern Recognition Letters* 136 (2020), 318–325. https://doi.org/10.1016/j.patrec.2019.10.013

[74] Stuart Reeves, Barry Brown, and Eric Laurier. 2009. Experts at Play: Understanding Skilled Expertise. *Games and Culture* 4, 3 (2009), 205–227. https://doi.org/10.1177/1555412009339730 arXiv:https://doi.org/10.1177/1555412009339730

[75] Alexander Renkl, Robin Stark, Hans Gruber, and Heinz Mandl. 1998. Learning from Worked-Out Examples: The Effects of Example Variability and Elicited Self-Explanations. *Contemporary Educational Psychology* 23, 1 (Jan. 1998), 90–108. https://doi.org/10.1006/ceps.1997.0959

[76] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135–1144.

[77] Justus Robertson, Athanasios Vasileios Kokkinakis, Jonathan Hook, Ben Kirman, Florian Block, Marian F Ursu, Sagarika Patra, Simon Demediuk, Anders Drachen, and Oluseyi Olarewaju. 2021. Wait, But Why?: Assessing Behavior Explanation Strategies for Real-Time Strategy Games. In *26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 32–42. https://doi.org/10.1145/3397481.3450699

[78] Margaret Salter and Gerald Klein. 2007. *After Action Reviews: Current Observations and Recommendations*. Technical Report. U.S. Army Research Institute for the Behavioral and Social Sciences.

[79] Taylor Lee Sawyer and Shad Deering. 2013. Adaptation of the US Army's After-Action Review for Simulation Debriefing in Healthcare. *Simulation in Healthcare* 8, 6 (Dec. 2013), 388–397. https://doi.org/10.1097/SIH.0b013e31829ac85c

[80] Morgan Klaus Scheuerman, Katta Spiel, Oliver L Haimson, Foad Hamidi, and Stacy M Branham. 2020. HCI guidelines for gender equity and inclusivity. *UMBC Faculty Collection* (2020). https://www.morgan-klaus.com/gender-guidelines.html

[81] Martin Schindler and Martin J Eppler. 2003. Harvesting project knowledge: a review of project learning methods and success factors. *International Journal of Project Management* 21, 3 (2003), 219–228. https://doi.org/10.1016/S0263-7863(02)00096-0

[82] David Schuler and Andreas Zeller. 2009. Javalanche: Efficient Mutation Testing for Java. In *Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering* (Amsterdam, The Netherlands) *(ESEC/FSE '09)*. Association for Computing Machinery, New York, NY, USA, 297–298. https://doi.org/10.1145/1595696.1595750

[83] Amber Shinsel, Todd Kulesza, Margaret M. Burnett, William Curran, Alex Groce, Simone Stumpf, and Weng-Keen Wong. 2011. Mini-crowdsourcing end-user assessment of intelligent assistants: A cost-benefit study. *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2011), 47–54.

[84] Dave Shreiner and The Khronos OpenGL ARB Working Group. 2009. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1* (7th ed.). Addison-Wesley Professional.

[85] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484.

[86] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144. https://doi.org/10.1126/science.aar6404 arXiv:https://science.sciencemag.org/content/362/6419/1140.full.pdf

[87] Richard Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick Pilarski, Adam White, and Doina Precup. 2011. Horde : A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction Categories and Subject Descriptors. *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems* 2.

[88] Hugues Talbot. 2000. WxPython, a GUI Toolkit. *Linux J.* 2000, 74es (June 2000), 5.

[89] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2013. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, Illinois, USA) *(KDD '13)*. Association for Computing Machinery, New York, NY, USA, 847–855. https://doi.org/10.1145/2487575.2487629

[90] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: Automated Testing of Deep-neural-network-driven Autonomous Cars. In *Proceedings of the 40th International Conference on Software Engineering* (Gothenburg, Sweden) *(ICSE '18)*. ACM, New York, NY, USA, 303–314. https://doi.org/10.1145/3180155.3180220

[91] Edward Tufte. 1990. *Envisioning Information.* Graphics Press, USA.

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

[93] U.S. Army. 1993. *Training Circular 25-20: A Leader's Guide to After-Action Reviews.* Technical Report. Department of the Army, Washington D.C., USA.

[94] Jasper van der Waa, Jurriaan van Diggelen, Karel van den Bosch, and Mark A. Neerincx. 2018. Contrastive Explanations for Reinforcement Learning in terms of Expected Consequences. *CoRR* abs/1807.08706 (2018). arXiv:1807.08706 http://arxiv.org/abs/1807.08706

[95] Bas van Opheusden, Gianni Galbiati, Zahy Bnaya, Yunqi Li, and Wei Ji Ma. 2017. A computational model for decision tree search. In *CogSci*.

[96] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354. https://doi.org/10.1038/s41586-019-1724-z

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

[98] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *ArXiv* abs/1711.00399 (2017).

[99] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 211 (Nov. 2019), 24 pages. https://doi.org/10.1145/3359313

[100] Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, StockholmsmÃ¤ssan, Stockholm Sweden, 5247–5256. http://proceedings.mlr.press/v80/weiss18a.html

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

[102] Bang Wong. 2011. Points of View: Color Blindness. *Nature Methods* 8 (May 2011), 441. https://doi.org/10.1038/nmeth.1618

[103] 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 IEEE RO-MAN 2017. *IEEE RO-MAN 2017.* http://opus.bath.ac.uk/55793/

[104] Robert K. Yin. 2008. *Case Study Research: Design and Methods (Applied Social Research Methods)* (fourth edition. ed.). Sage Publications. http://www.amazon.de/Case-Study-Research-Methods-Applied/dp/1412960991%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1412960991

[105] Tom Zahavy, Nir Ben Zrihem, and Shie Mannor. 2016. Graying the black box: Understanding DQNs. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48* (New York, NY, USA) *(ICML'16)*. JMLR.org, 1899–1908. http://dl.acm.org/citation.cfm?id=3045390.3045591

[106] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. 2016. DeepRED – Rule Extraction from Deep Neural Networks. In *Discovery Science*, Toon Calders, Michelangelo Ceci, and Donato Malerba (Eds.). Springer International Publishing, Cham, 457–473.