AN ABSTRACT OF THE DISSERTATION OF

Jonathan E. Dodge for the degree of Doctor of Philosophy in Computer Science presented on March 28, 2022.

Title: Explanations and Processes to Enable Humans to Assess AI with Respect to Manipulable Properties

Abstract approved: _____

Margaret Burnett

Assessing AI systems is difficult. Humans rely on AI systems in increasing ways, both visible and invisible, meaning a variety of stakeholders need a variety of assessment tools (e.g., a professional auditor, a developer, and an end user all have different needs). We posit that it is possible to provide explanations and assessment processes that enable AI non-experts observing multiple intelligent agents in sequential domains to differentiate the agents with respect to a property (e.g., quality or fairness), as well as articulate justification for their differentiation. Further, we hypothesize that if the property can be manipulated in a highly controllable fashion, then it is possible to *measure* the quality of an explanation and/or assessment process by its ability to expose that such manipulation has occurred. This dissertation presents our contributions in explanations, processes, and manipulations for assessment. Specifically, we present our investigations into explanations to judge fairness of a classifier, the After-Action Review for AI process to structure explanation consumption, the Ranking task for explanation evaluation, and The Mutant Agent Generation approach for introducing controllable variation. By improving explainability of AI in all these phases, we seek to empower assessors to calibrate trust in the system appropriately.

Explanations and Processes to Enable Humans to Assess AI with Respect to Manipulable Properties

by

Jonathan E. Dodge

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented March 28, 2022
Commencement June 2022

Doctor of Philosophy dissertation of Jonathan E. Dodge presented on March 28, 2022.

APPROVED:

Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Jonathan E. Dodge, Author

ACKNOWLEDGEMENTS

I'd like to thank those who supported me along the way; they are too many to name, but here are a few: *Margaret Burnett*, my PhD advisor who invested the second most hours into this document after me; *Ron Metoyer*, my MS advisor who taught me a great deal before starting on this dissertation; *Diana Paris*, my partner who kept me alive as I worked too hard; *Cleon* and *Marcia Dodge*, my parents who taught me that learning is a great pastime; *Andrew Anderson*, my colleague in PhD trials and tribulations who kept us on the proper statistical path; *Alan Fern*, who wrangled grant funders and taught us all about AI; *Tom Dietterich*, *Fuxin Li*, and *Prasad Tadepalli*, my committee members who helped sharpen this dissertation with important feedback; *Q. Vera Liao*, *Yunfeng Zhang*, *Rachel K. E. Bellamy*, and *Casey Dugan*, for their invaluable support during and after my IBM internship; *Jed Irvine*, for excellently underpinning the development work for several studies; *Rupika Dikkala* and *Roli Khanna*, for performed the critically important task of running several studies; *Matthew Olson*, who found key problems in my original agent design; *Minsuk Kahng*, for improving the visualizations in our interfaces; *Sean Penney*, *Kin-ho Lam*, *Zhengxian Lin*, *Theresa Mai*, *Sai Raja*, *Caleb Matthews*, *Nicholas Kiddle*, *Evan Newman*, *Christopher Perdriau*, *Zeyad Shureih*, *Teresita Carolina Guzman Nader*, *Chimdi Chikezie*, *Yashwanthi Anand*, *Delyar Tabatabai*, *Anita Ruangrotsakun*, *Larissa Letaw*, and *Sabyatha Sathish Kumar*, for staffing various projects without which we could not have proceeded; *Kevin McGrath*, for keeping me in the PhD program; *Forrest Briggs*, *Nels Oscar*, *Iftekhar Ahmed*, *Jonathan Palacios*, and *Harish Dayapule*, for their assistance with class projects and influencing my thinking about CS; and the *grant funders*, *professors and teachers*, *burrito purveyors*, and *anyone else not listed above* who positively impacted my life.

This work was supported by DARPA #N66001-17-2-4030 and NSF #1314384. This research was sponsored in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of: DARPA, NSF, US Army Research Office, US Government, UK Ministry of Defence or U.K. Government.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure | Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF APPENDIX FIGURES

Figure Page

LIST OF APPENDIX FIGURES (Continued)

LIST OF APPENDIX TABLES

LIST OF APPENDIX TABLES (Continued)

Table Page

Chapter 1: General Introduction

Artificial Intelligence (AI) is becoming more pervasive in society; from automatic news feeds to medical diagnoses to self-driving cars. Given the importance of the decisions these systems make, humans need the ability to discern if the AI is reliably making good decisions for the right reasons to decide on acceptance/rejection. In some cases, public rejection has been severe enough as to cause people to even attack self-driving cars with rocks and knives [126]). To avoid outcomes such as this, Explainable Artificial Intelligence (XAI) strives to make AI systems more transparent, meaning that the reasons for decisions are clear [169]. The demand for XAI is partly driven by GDPR regulations that grant data subjects the right to know about,

> *...the existence of automated decision-making... [and] meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing.* [53]

However, exposing "meaningful information" about complicated systems poses many challenges, which grow as the domain or decision making process increases in complexity.

1.1 Supporting *Which* XAI users?

Our goal for exposing meaningful information is to support assessment, which first raises questions of *what* that means and *who* is conducting such assessment. We consider assessment to be the process by which some human(s) make judgments about properties of a system (e.g., to determine *"if it is fit for the purpose"* [61]). We take a broad view of assessment by including a wide variety of tasks, such as sensemaking and fault localization. We envision a wide variety of interested parties conducting such assessment: end users, auditors, judges, professional developers, managers, designers, etc. Thus, XAI designers need to support ways diverse individuals go about diverse tasks in an interactive XAI system. Ultimately, *anyone* could seek to assess a system when its decisions are very important to them: from diverse devs, managers, designers working on a deployment (as in [39]); to end users for whom this agent's decisions are critical

(e.g., medical caregivers, internet of things users). According to the ACM Code of Ethics 1.1, *"all people are stakeholders in computing."* Our goal is not for *every* user to do this, but to create the tools and techniques so that *any* user who needs to do this would be able to succeed.

In general, assessors might want to inspect a wide variety of properties, such as strength or speed. In this dissertation, we will focus on two ordinal properties: (a) *Fairness* of a Decision Support System in Chapters 2 and 3; and (b) *Ability to win a game* in Chapters 4, 5, 6, and 7.

Having specified the properties of interest, we must specify a setting in which the user will assess AI. XAI research has demonstrated that explaining "one-shot" decisions has proven to be a significant challenge, with much recent research in that space (e.g. [22]). Making decisions in *sequential* domains is even more difficult, as a decision's dependence on previous decisions (and possible future decisions) should be made clear. We embrace this challenge, as sequential domains encompass important problems, such as planning. In this dissertation we first examine a domain with "one-shot" or "episodic" decisions [155] by discussing recidivism prediction in Chapters 2 and 3. We then move to *sequential* decisions in two different domains: Tug-of-War (Chapters 4 and 5) and MNK Games (Chapters 6 and 7.

Regardless of the domain, people assess AI with some goal in mind. One option is the traditional "acceptance test" [61], whose goal is to decide whether or not to use *one* particular system. A different goal might be to compare two or more agents to find the most suitable one. For example, before the final agent could be acceptance tested (e.g., by an external auditor or an HCI study), developers often create many agent configurations as part of the *hyperparameter tuning* process (see Section 11.4 of Goodfellow et al. [56]). Selecting from among these agents is akin to the ranking task we propose, and so our XAI tools might serve these developers. Similarly, another kind of assessor, an end user, may need to determine in which situations, among many, a single agent is performing "well enough" for them to choose to deploy the agent—which can considered a higher granularity of acceptance test.

During formative studies (e.g., [136, 47, 11]) we realized that participants' cognitive energy was being sapped by having to create not *only* a mental model, but also a *process* by which to create that mental model. This realization led us to create such a process (which we term After-Action Review for AI or "AAR/AI") for them and integrate it

into explanation environments as detailed in Chapters 4 and 5.

1.2 Supporting XAI Researchers

Alongside supporting assessors, we also would like to support *XAI researchers*. In particular, having created an XAI system, it remains difficult for such researchers to measure the extent to which that system is succeeding (or not). This leads us to the idea of **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"*.

To measure resolution, we sought to improve upon current empirical XAI tasks (identified by Hoffman et al. [69]) by introducing the Ranking Task, discussed in Chapter 6. The Ranking task takes N agents as input and the assessor puts them in order based on the desired property.

To induce controllable differences in the AI being assessed to see if the explanation can resolve such differences, we introduced a novel empirical strategy in Chapter 6. Our approach is called "Mutant Agent Generation," and it works by adding noise to the neural network weights. The noise is parameterized and targeted such that we can control the extent of the damage to the network. Our approach is conceptually similar to the idea of Mutation Testing [38] in Software Engineering, in which one can determine the quality of a test suite by providing it many copies of mutant source code with random changes (e.g., swapping a '+' for '-') and then measuring how many of these mutants are caught. In our case, we treat the weights of a neural network as analogous to "source code" in which we intend to reveal the presence (or absence) of mutation, possibly via explanation. This provides XAI researchers the ability to evaluate in a controlled fashion which assessment techniques (e.g., humans inspecting explanation A, humans inspecting explanation B, an automated system, etc.) are better than others by their ability to find mutants.

1.3 Thesis Statement

Our thesis statement in two parts:

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

- **Part 1:** It is possible to provide explanations and assessment processes that enable AI non-experts observing multiple intelligent agents in sequential domains to differentiate the agents with respect to a property (e.g., quality or fairness), as well as to articulate the reasons for their differentiation.
- **Part 2:** If the property can be manipulated in a highly controllable fashion, then it is possible to *measure* the quality of an explanation and/or assessment process by its ability to expose that such manipulation has occurred. In particular, an explanation with *higher resolution* will be able to expose *smaller manipulations*.

Chapter 2: Explaining Models: An Empirical Study of How Explanations Impact Fairness Judgment

By: Jonathan Dodge, Q. Vera Liao, Yunfeng Zhang, Rachel K. E. Bellamy, and Casey Dugan

Appeared in: 24th International Conference on Intelligent User Interfaces (IUI '19). ACM, New York, NY, USA, 275–285.
DOI: https://doi.org/10.1145/3301275.3302310

Abstract: Ensuring fairness of machine learning systems is a human-in-the-loop process. It relies on developers, users, and the general public to identify fairness problems and make improvements. To facilitate the process we need effective, unbiased, and user-friendly explanations that people can confidently rely on. Towards that end, we conducted an empirical study with four types of programmatically generated explanations to understand how they impact people's fairness judgments of ML systems. With an experiment involving more than 160 Mechanical Turk workers, we show that: 1) Certain explanations are considered inherently less fair, while others can enhance people's confidence in the fairness of the algorithm; 2) Different fairness problems–such as model-wide fairness issues versus case-specific fairness discrepancies–may be more effectively exposed through different styles of explanation; 3) Individual differences, including prior positions and judgment criteria of algorithmic fairness, impact how people react to different styles of explanation. We conclude with a discussion on providing personalized and adaptive explanations to support fairness judgments of ML systems.

2.1 Introduction

Increasingly, important decisions that impact human lives and societal progress are supported by machine learning (ML) systems. Examples where ML systems are used to make decisions include hiring, marketing, medical diagnosis, and criminal justice. This trend gives rise to concerns about algorithm fairness—or possible discriminatory consequences for certain groups of individuals. Machine learning algorithms are trained based on data from past decisions, decisions which may have themselves been biased and discriminatory. Research shows that by optimizing for the unitary goal of accuracy, ML algorithms trained on historical data not only replicate, but may amplify existing biases or discrimination [209]. The possibility of spiraling discriminatory consequences is driving a distrust and "fear of AI" in public discussions (e.g., [2, 1]).

There is a growing body of work on developing non-discriminatory ML algorithms (e.g., [82, 78, 206]), equal attention has not been paid to the human scrutiny necessary to identify and remedy fairness issues. The need for such research is highlighted by recent studies, which show that algorithmic fairness often may not be prescriptively defined, but is multi-dimensional and context-dependent [58]. Public scrutiny of the application of risk assessment algorithms in the criminal justice system [3, 103] brings attention to the need to improve the accountability and fairness of such algorithms.

Accurately identifying fairness issues in ML systems is extremely challenging, however. Most ML algorithms aim to produce only prediction or decision outcomes, while humans tend to rely on information about decision-making *processes* to justify the decisions made. ML algorithms are often seen as "black boxes", where one can only see the output and make a best guess about the underlying mechanisms. This problem is further exacerbated by the popularity of deep learning algorithms, which are often unintelligible even for experts. This lack of transparency drives a sweeping call for *explainable artificial intelligence* (XAI) in industry, academia, and public regulation. For example, the EU General Data Protection Regulation (GDPR) requires organizations deploying ML systems to provide affected individuals with meaningful information about the logic behind their outputs.

Critically, explanations are not just for people to understand the ML system; they also provide a more effective interface for the human in-the-loop, enabling people to identify and address fairness and other issues. When people trust the explanation, it

follows that they would be more likely to trust the underlying ML systems.

Much research is dedicated to generating explanations in various styles, including model-agnostic approaches [149, 112] applicable to any ML algorithm. However, this body of research is criticized for "approaching this [XAI] challenge in a vacuum considering only the computational problems" [118] without the quintessential understanding of how *people* perceive and use the explanations.

In this paper, we conduct an empirical study on how people make fairness judgments of ML systems and how explanation impacts that judgment. We aim to highlight the nuances of such judgments, where there are different types of fairness issues, different styles of explanation, and individual differences, to encourage future research to take more user-centric and personalized approaches.

Specifically, we identify four styles of explanation based on prior XAI work and automatically generate them for a ML model trained on a real-world data set. In the experiment, we explore the effectiveness of explanations in exposing two types of fairness issues: model-wide unfairness produced by biased data and fairness discrepancies in cases from different regions of the feature space. Our user study demonstrates that judging fairness is not only influenced by explanation design, but also an individual's prior position on algorithmic fairness, including both the general trust of ML systems for decision support and one's position on using a particular feature. We also present user feedback for the four styles of explanation. Our results provide insights on the mechanisms of people's fairness judgment of ML systems and suggest design guidelines for explanations to facilitate fairness judgment making. We first review relevant work, then present the study overview and research questions.

2.2 Background

2.2.1 Fairness of Machine Learning Systems

One of several definitions for algorithmic fairness is: "*...discrimination is considered to be present if for two individuals that have the same characteristic relevant to the decision making and differ only in the sensitive attribute (e.g., gender/race) a model results in different decisions*" [24]. The consequence of deploying unfair ML systems could be *disparate impact*, practices which adversely affect people of one protected characteristic

more than another in a comparable situation [24, 58].

Despite the "statistical rationality" of ML techniques, it has been widely recognized that they can lead to discrimination. Many reasons can contribute to this, including biased sampling, incorrect labeling (especially with subjective labeling), biased representation (e.g., incomplete or correlated features), suboptimal or insensitive optimization algorithm, shift of population or data distribution, and failure to consider domain-specific, legal, or ethical constraints [24, 60]. Various techniques have been proposed to address these causes of "unfair algorithms" [82, 60, 206, 208, 78]. For example Calders and Žliobaitė suggested techniques to de-bias data [24], including modifying labels of the training data, duplicating or deleting instances, adding synthetic instances, and transforming data into a new representation space.

We use a recently proposed data de-biasing method that applies a preprocessor to transform the data [25]. The result is a new dataset which is "fairer"–while also limiting local deformations from the data transformation. This is because the preprocessor optimizes data transformations with respect to penalties that rise with the magnitude of a feature change (e.g., changing a person's age from 5 to 60 will result in a higher penalty than from 5 to 8). Simply put, if raw data contains biases that lead to an unfair model with a discriminatory feature (e.g., certain racial category is weighed more negatively than others), the data preprocessing mitigates the bias introduced by that feature. This method has the benefit of retaining all features (as opposed to removing the discriminatory feature), which, among other benefits, would also allow exploration of correlations among them [24].

The above debiasing techniques are *normative* by nature, i.e., they rely on prescriptively defining the criteria of fairness in order to optimize for those criteria. A recent paper pursued a complementary *descriptive* approach by empirically studying how people judge the fairness of features used by a decision support system in the criminal justice system [58]. Their study uncovered the underlying dimensions in people's reasoning of algorithmic fairness, and demonstrated individuals' variations on these dimensions.

We adopt the same descriptive view, empirically studying how people judge fairness of an ML system and considering individual differences in their prior position on algorithmic fairness. However, we also fill a gap in prior work by investigating how normative fairness (via the use of the preprocessor) is *perceived* by people, and what factors impact such perception.

2.2.2 Explanation of Machine Learning

Explainable AI (XAI) is a field broadly concerned with making AI systems more transparent so people can confidently trust an AI system and accurately troubleshoot it — fairness issues included. Work on model explanations can be traced to early work on expert systems [174, 30], which often explicitly revealed reasoning rules to end-users. There has been a recent resurgence of XAI work driven by the challenge to interpret increasingly complex ML models, such as multi-layered neural networks, and by the evidence that ethical concerns and lack of trust hampers adoption of AI applications [104, 60].

A large volume of XAI work is on producing more interpretable models while maintaining high-level performance (e.g. [28, 105]), or on methods to automatically generate explanations. Given the complexity of current ML models, explanations are often *pedagogical* [181], meaning that they reveal information about how the model works without faithfully representing the algorithms. Many methods rely on some kind of sensitivity analysis to illustrate how a feature contributes to the model prediction [149, 112], so they can be model-agnostic, thus applicable to complex models. For example, LIME explains feature contribution by what happens to the prediction when the feature values change (perturbing data) [149]. Another common category is *case-based explanations*, which use instances in the dataset to explain the behavior of ML models. Examples include using counter-examples [193] and similar prototypes from the training data [89]. Case-based explanations are considered easy to consume and effective in *justifying* the decision, but may be insufficient to explain how the model works.

Work on how people perceive explanations of ML systems is a growing area [171, 109, 5, 98] which aims to inform the choices and design of explanations for particular systems or tasks. Recent work calls for taxonomic organizations of explanations to enable design guidelines [109]. In earlier work on explaining expert systems, researchers argued the difference between *description v.s. justification* by making not only the *how* visible to users, but also the *why* [174]. Accordingly, Wick and Thompson discussed the taxonomy of *global-local* explanations [199]. During initial practice, users may need global explanations that describe "how the system works." When interacting with a deployed system, users tend to rely on justifications of why the system did what it did on particular cases.

Another useful taxonomy is proposed by Kulesza et al. [98] by considering two di-

mensions of explanation fidelity: *soundness* (how truthful each element in an explanation is with respect to the underlying system) and *completeness* (the extent to which an explanation describes all of the underlying system). They empirically showed that the best mental models arose from explanations with both high completeness and high soundness. However, crafting highly complete explanations comes with a tradeoff, as completeness usually requires increasing the length and complexity of the explanation, which was shown to be detrimental to task performance and user experience in previous studies [123].

While researchers have explored user preferences in explanation styles, they have paid little attention to individual differences in such preferences. Meanwhile, psychological research has long been interested in individual differences in explanatory reasoning. For example, research shows that some prefer simple, superficial explanations and others are more deliberative and reflective in their reasoning [50, 93]. Such individual differences can be predicted by cognitive style (e.g., cognitive reflection, need for cognition) [50] and culture [93]. It is therefore possible that individuals differ in preferences for completeness and soundness of explanations.

Our work is concerned with how explanations impact fairness judgments of ML systems. We build on a recent study by Binns et al. [16] that examined human perception of a classifier's fairness in the insurance domain. They provided four different explanation types applied to *fictional scenarios* to elicit fairness judgments. While the study provided rich qualitative insights on the heuristics people use to make fairness judgments, the authors acknowledge a lack of ecological validity, as the explanations were not drawn from real ML model output. Moreover, the explanations were not produced for the same data points, so they were incommensurate, which could possibly explain the absence of conclusive preference in their quantitative results.

Our work set out to overcome limitations on prior work by *automatically generating* four types of explanations on a real ML model and quantitatively examining how they impact people's fairness judgments. Combining this advancement with the use of the data preprocessor allowed us to perform more carefully-controlled experiments for ML fairness perception than prior work.

2.3 Study Overview

Related work informed four main considerations of our study: use case, choices of explanation styles, fairness issues we focus on, and the individual differences we explore. Through both quantitative and qualitative results, we aim to answer the following RQs:

RQ1 How do different styles of explanation impact fairness judgment of a ML system?

RQ1a Are some explanations judged to be fairer?

RQ1b Are some explanations more effective in surfacing unfairness in the model?

RQ1c Are some explanations more effective in surfacing fairness discrepancies in different cases?

RQ2 How do individual factors in cognitive style and prior position on algorithmic fairness impact the fairness judgment with regard to different explanations?

RQ3 What are the benefits and drawbacks of different explanations in supporting fairness judgment of ML systems?

2.3.1 Use Case: COMPAS recidivism data

We conducted an empirical study with a ML model trained on a real data set. Like Grgic-Hlaca et al. [58], we chose a publicly available data set for predicting risk of recidivism (reoffending) with known racial bias[1]. The data set was collected in Broward County, Florida over a two year span. It is used by COMPAS (Correctional Offender Management Profiling for Alternative Sanctions), a commercial algorithm to help judges score criminal defendants' likelihood of reoffending. However, ProPublica has reported on troubling issues with the COMPAS system [103, 3]. First, the classifier may have low overall accuracy ([3] reported 63.6%). Second, the model is reported to exhibit racial discrimination, with African American defendants' risk frequently overestimated.

We chose a criminal justice use case because it carries weight to elicit reaction on fairness, even for the general population. Note our goal is not to study the actual users of COMPAS. Rather, we are using the use case as a "probe" to empirically study fairness judgments. The same use case was analyzed in previous studies to understand how people perceive algorithmic fairness with regard to features used [58].

[1] `https://www.kaggle.com/danofer/compass`

2.3.2 Explanation Styles

We chose to programmatically generate the four types of explanations introduced by Binns et al. [16] (details to be discussed in System Overview section) because they represent a set of common approaches in recent XAI work. They embody the categorization of *global vs. local* explanations. Specifically, influence and demographic-based explanations are global styles as they describe how the model works; sensitivity and case-based explanations are local styles, as they attempt to justify the decision for a specific case. These explanation styles vary along the taxonomy introduced by previous work in other ways: e.g., sensitivity based explanation is similar to a "what if" [109] and the case-based explanation is the least "sound" of the explanation types discussed [98].

2.3.3 Fairness Issues

Given our use case, we consider fairness issues in terms of racial discrimination. While there are other controversial features in the dataset [58], race is generally considered inappropriate to use in predicting criminal risk (termed *protected variable*). We focus on two types of fairness issues.

2.3.3.1 Model Unfairness

As discussed, the COMPAS data set is known to be racially biased, but we mitigate that bias by using the data processing method in Calmon et al. [25]. In the experiment, we employ a **data processing** technique as a between-subject variable. By comparing participants' fairness judgment for a model trained on the *raw* data to one trained on *processed* data, we aim to understand whether participants could identify the model-wide fairness issue and whether certain explanations expose the problem better.

2.3.3.2 Case-specific disparate impact

Predictions from an ML algorithm are not uniformly fair–consider *disparate impact* from a protected variable. For example, if two individuals with identical profile features but different racial categories receive different predictions, those predictions should be considered unfair [24, 58]. Statistically, these cases are on the decision boundary of

the feature space given the relatively small weight of the race factor, meaning they have low-confidence predictions that may be unfair. In the experiment, we introduce **disparate impact** by race factor as a within-subject variable (i.e., each participant would be asked to judge some cases with disparate impact and some not). We adopt a factorial design with *disparate impact* and *data processing*. For subjects given models after data processing [25], disparate impact is reduced. We aim to discover how well participants identify the case-specific fairness issues when given different explanations.

Our hypothesis is that because local explanations focus on justifying a particular case, they should more effectively surface fairness discrepancies between cases. In contrast, global explanations may require additional effort to reason about the position of the case with respect to the decision boundary (e.g., "*This person's features all have no impact in the model, except race*"). Note that local explanations may expose the case-specific fairness issue differently. Case-based explanation exposes the boundary position with a low percentage of cases justifying the decision. Sensitivity-based explanation explicitly describes disparate impact–"*Changing this person's race changes the prediction*".

2.3.4 Individual Difference factors

Based on prior work, we focus on two areas of individual factors: cognitive style and prior position on algorithmic fairness. For cognitive style, we measure the individual's *need for cognition* [21]. For prior positions, we consider two levels: *general position on the fairness of using ML systems* for decision support and *position on the fairness of using a particular feature*. Here we focus on the race factor.

2.4 System Overview

2.4.1 Re-offending Prediction Classifier

The model is a binary classifier predicting whether an individual in the COMPAS data set is likely to re-offend or not, implemented by Scikit-learn's logistic regression. The use of a regression model is ecologically valid—many current decision support systems use such simple and interpretable models [189]. However, the explanation styles we study are not limited to regression models.

We built the model using a subset of features in the COMPAS dataset[2], including *Race* as the feature with fairness issues. For simplicity, we focus on two racial groups (Caucasian and African-American), and filtered out the others. Other features included: *Age* (18-29/30-39/40-49/50-59/>59), *Charge Degree* (Felony/Misdemeanor), *Number of Prior Convictions* (0/1-3/4-6/7-10/>10), and *Had Juvenile Convictions* (True/False). According to Grgic-Hlaca et al. [58], charge degrees and criminal history were deemed fair in a similar use case. Age is also important in assessing re-offense risk. Following statistical convention, all categorical features are dummy coded using the median category[3] as the reference level, where possible.

The accuracy of the model is 67.1% on raw data and 67.6% on processed data, comparable to reports on the accuracy of COMPAS system [18, 103]. Note that logistic regression also produces a confidence level in the form of predicted class probabilities.

2.4.1.1 Data processing and cases with disparate impact

We used the method introduced in [25] to perform data processing and then re-trained the model. The resulting model reduced bias against the African American group, as evidenced by the feature co-efficient being reduced from 0.177 to -0.036. (A feature co-efficient of 0 corresponds to the feature having no effect in the decision.)

To identify cases with unfair treatment of disparate impact, we follow the definition "*treating one person less favorably on a forbidden ground than another...in a comparable situation*" [24]. That is, if perturbing a test example's protected variable (race) changes the algorithm's prediction, we consider it to have disparate impact. We found 23 cases of disparate impact using the raw classifier on the raw dataset—all very near to the decision boundary[4].

[2]We split the data into training set (4222 samples) and testing set (1056).

[3]Meaning after we had divided numerical "age" into binary "age_18-29", "age_30-39", "age_40-49", "age_50-59", and "age_>59", we would drop the median (here, "age_40-49"), because it is redundant and can be inferred by process of elimination on the other features.

[4]The raw data classifier's confidence on the disparately impacted sample group had an average of 52% and a max of 54%. The processed data classifier had average and max confidence both at 50%.

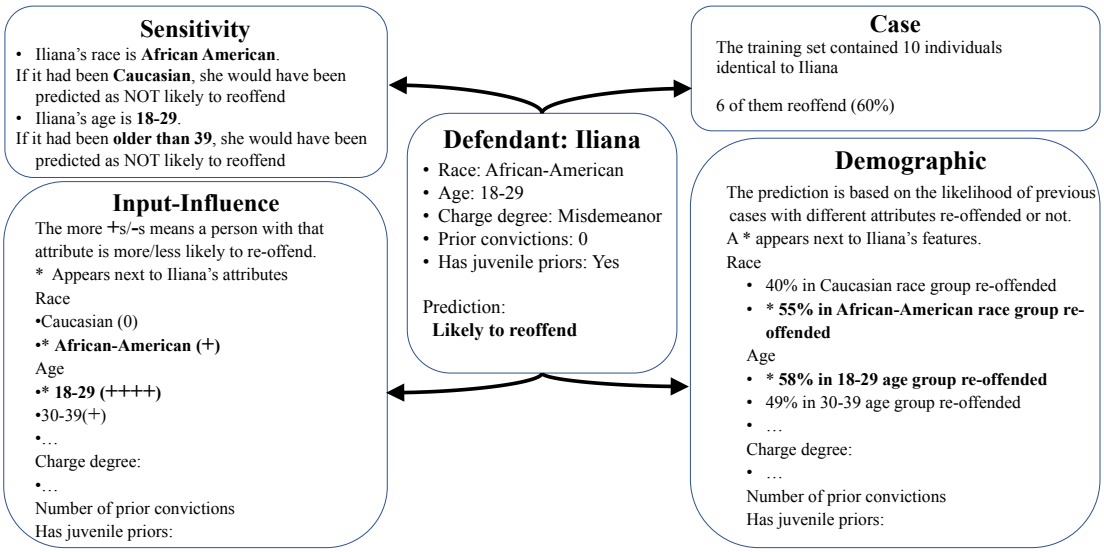


Figure 2.1: Examples of explanations generated using the raw data classifier, adjusted and truncated for brevity. Consult our supplementary materials for full explanation output from both classifiers – as seen by participants (including * and highlights).

2.4.1.2 Sampling cases for the user study

Due to user study time constraints, we could only show each user a small sample of the explanations. Since we intended to study fairness discrepancies between disparately impacted and non-impacted cases, we over-sampled the former category. Among the 23 disparately impacted instances, we sampled all 8 unique cases (i.e. the rest had the same feature-values as one of the 8 we sampled). From the non-impacted group of 992 instances, we sampled 16 unique cases.

2.4.2 Explanation Generation

As discussed, we patterned our explanations, shown in Figure 2.1 (truncated version, see supplementary materials for the full version), after the templates presented by Binns et al. [16]. While Binns et al. manually created examples of these explanation, we developed programs to automatically generate them to obtain comparable explanation versions for the same data point, controlling for differences in representation and presentation. These generation methods can also be broadly applied to ML prediction models using relational features.

2.4.2.1 Input Influence-based Explanation

An Input Influence-based Explanation describes the decision boundary itself. Because the feature coefficients of the logistic regression model encode the relative importance of each feature, we present them as strings of '+' and '-' in our explanations, as shown in Figure 2.1. To do this, we discretized them into 11 buckets, based on the range of the maximum and minimum coefficient. This type of explanation is *global*, since the decision boundary is a property of the classifier, and thus is described the same way for all samples.

2.4.2.2 Demographic-based Explanation

A Demographic-based Explanation describes the structure of the training data and how it is distributed with respect to the decision boundary. We simply summarize, for the training data matching each feature category[5], the percentage of data points with the same label as predicted for the presented example. This type of explanation is *global* and generates the same description for all samples on each side of the decision boundary.

2.4.2.3 Sensitivity-based Explanation

A Sensitivity-based Explanation seeks to modify the presented example along each feature until the prediction changes. When the prediction does change, we report back to the user the necessary feature change to produce the change in output. This type of explanation is *local*, as it is specific to each presented example, and it justifies the decision by indicating changes needed to produce a different output.

2.4.2.4 Case-based Explanation

To construct a case-based explanation, we perform a nearest neighbor search in the training data to find cases similar to the presented example. Since our study has a large data set with respect to size of the feature space, we frequently find neighbors occupying

[5]We formed feature categories when we divided numerical features into binary indicators variables via discretization. For example, the numerical feature "age" became the following category of binary features: "age_18-29", "age_30-39", "age_40-49", "age_50-59", and "age_>59".

the same feature space location as the sample presented for explanation. In this situation, we show the % of those neighbors with the same label as the prediction. When no exact matches are found, we show the features and label for the nearest neighbor in the training data. This is a modification to the design in Binns et al., which describes only a single identical or similar case. This explanation is *local*, and it attempts to justify the decision by indicating similar examples with similar outputs.

2.5 Methodology

Our study adopted a mixed design by having data processing (raw or processed) and explanation styles (4 styles) as between-subject variables, and disparate impact as a within-subject variable. Each participant completed 6 fairness judgment trials in a random order, where each trial consisted of judging a single case in the test data. From the 8 disparately impacted cases we randomly selected 2 trials, and the remaining 4 trials from the 16 non-impacted cases.

In September 2018, we recruited 160 Amazon Mechanical Turk workers, with the criteria that the worker must live in the US and have completed more than 1000 tasks with at least a 98% approval rate. We randomly assigned participants to the 8 conditions (2 data processing treatments $\times$ 4 explanations). Among them, 62.5% were male, 78.8% were self-identified as Caucasian, 29.4% were under 30, and 13.3% were above 50. In our quantitative analysis, we included participants' race and gender as covariates but observed no significant effect.

2.5.1 Study Procedure

We conducted an online survey-style study. Participants first provided informed consent, then reviewed a scenario about a ML system developed to assess defendants' re-offense risk, to help a judge make bail decisions. Each participant was given 6 trials, and within each trial, the participant would first review information about an individual (Figure 2.1, center region), then make his or her own prediction about whether that individual would re-offend. This step was to ensure they carefully considered the features used in the prediction. Subsequently, they would be presented with the prediction from the machine learning classifier with one of the four styles of explanations. They were asked to rate

agreement with: "*How the software made the prediction was fair*" based on a 1 (Strongly Disagree) to 7 (Strongly Agree) Likert Scale, together with an open question justifying their rating.

We checked attention after the third trial, by asking participants to find the feature not used by the software from a list of choices. All participants passed the check. After completing all tasks, participants would answer a survey measuring their individual differences and collecting demographic data. On average, the study took 18 min to complete, and each participant was compensated with $3.

2.5.2 Individual Differences

We measured two types of individual differences: prior positions on algorithmic fairness and cognitive style. For prior positions, we further differentiated between one's general position on using ML systems to assist decision-making and one's position on the fairness of using race as a feature. We used a semantic differential scale that is often used to measure attitude on controversial topics [106]. Specifically, we asked participants to rate the statements "*Considering race as a factor in criminal risk prediction*" and "*Using machine learning software to replace or augment human decision making*", on 5 pairs of bipolar adjectives such as "*harmful-beneficial*" and "*unethical-ethical*" with a 7-point Likert scale. The ratings of the 5 pairs were averaged to represent the individual's prior position. The Cronbach Alpha for the scale on general ML position is 0.95, and for the race feature is 0.98, showing excellent consistency. For cognitive style, we chose to measure one's Need for Cognition using only 4 of the items from the standard scale [21] due to the time constraints.

2.6 Results: Quantitative

We start with examining the effect of explanation style, data processing (*raw/processed*) and disparate impact (*true/false*) on participants' fairness judgment (RQ1). We then explore how individual differences, including prior position on fairness of ML, prior position on fairness of using the race feature, and need for cognition, affected the judgment (RQ2). We performed statistical analysis in R using the `lmerTest` package to run mixed-model regressions.

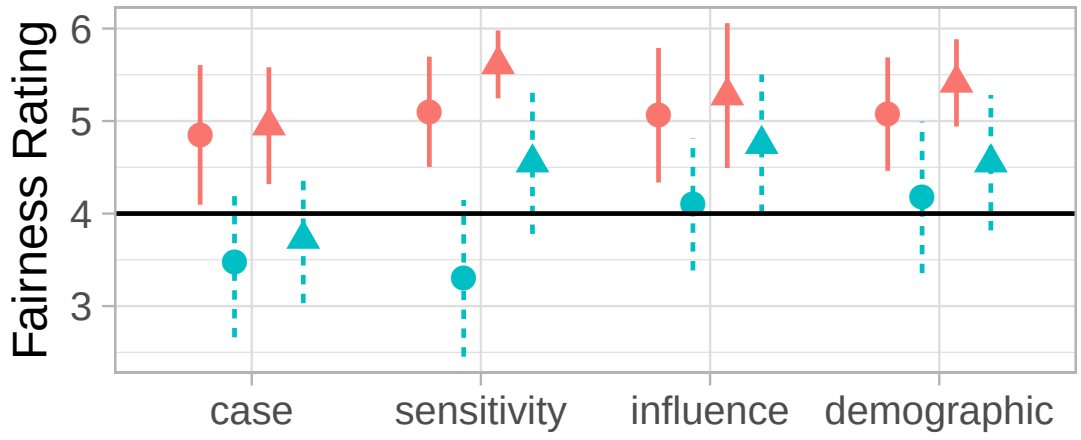


Figure 2.2: Overall mean ratings of fairness, per explanation type, data process treatment *(raw=●, processed=▲)*, and sample group *(impacted=blue dashed lines, non-impacted=red solid lines)*. The lines indicate the 95% confidence intervals. The thick black line at represents the middle of the Likert scale from 1–7.

2.6.1 Explanation, data processing, and disparate impact

Given the complexity of the statistical model, we first describe the trends with the descriptive data, then report statistical testing results. In Figure 2.2, we plot the mean and the 95% confidence interval of the mean of fairness ratings in all experiment treatments, showing several trends:

1. Predictions made on the processed data (triangles) were rated fairer than those on the raw data (circles). It suggests that participants perceived fairness issues for the model trained on the raw data, and the processing technique mitigated the problem.
2. Predictions made on cases with disparate impact (blue dashed lines) were rated less fair than those without it (red solid lines). This shows participants' fairness perceptions align with the presence of a fairness discrepancy between groups.
3. Explanation styles made nuanced differences. As expected, the two local explanations led to higher discrepancy of fairness ratings between disparately impacted cases and non-impacted cases (difference between the dashed and solid lines) than the two global explanations. Thus, the former are more effective in exposing case-specific fairness issues. Moreover, this difference is most prominent for sensitivity-based explanations applied to raw data. This could be caused by sensitivity-based explanation

being the most explicit in exposing disparate impact, while data processing mitigated the problem.

We now report the statistical significance of these observed trends. In particular, to validate that sensitivity-based explanation is most effective in exposing the disparate impact issue in the raw data, we expect to see a three-way interaction between explanation style, data processing, and disparate impact. We construct a mixed-effect regression model with the three-way interaction (and all the lower order interactions) as fixed effects, and participant as a random effect. We controlled for participants' gender and race as covariates and neither had a significant effect. The three-way interaction we expected is *not* significant, $F(3, 152) = 0.54$, $p = 0.66$. There is a marginally significant[6] two-way interaction between explanation style and disparate impact, $F(3, 152) = 2.35$, $p = 0.07$, and significant main effect of disparate impact, $F(1, 152) = 103.25$, $p < 0.001$, and data processing, $F(1, 152) = 4.65$, $p = 0.03$.

The main effect of data processing and disparate impact prove statistical significance for the first two observed trends. The two-way interaction indicates that explanation styles had differential impact on exposing the disparate impact issue. We conduct pairwise comparison for this interactive effect to identify pairs of explanation styles where this perceived fairness discrepancy differs significantly. We found that if we use sensitivity-based explanation as the reference level, influence-based explanation is significantly different, $F(1, 156) = 5.14$, $p = 0.02$, and demographic-based explanation is marginally significant, $F(1, 156) = 3.29$, $p = 0.07$; If we use case-based explanation as the reference, influence-based explanation is marginally different, $F(1, 156) = 3.36$, $p = 0.07$. This validates the observation that local explanations are more effective than global ones in exposing fairness discrepancies in different cases.

While we did not find statistical significance of the three-way interaction that validates the effectiveness of sensitivity-based explanation, a possibility is that there are individual differences for which the model did not account. In the next section, we explore that possibility.

[6] We consider $p < 0.05$ as significant, and $0.05 \leq p < 0.10$ as marginally significant, following statistical convention [35]

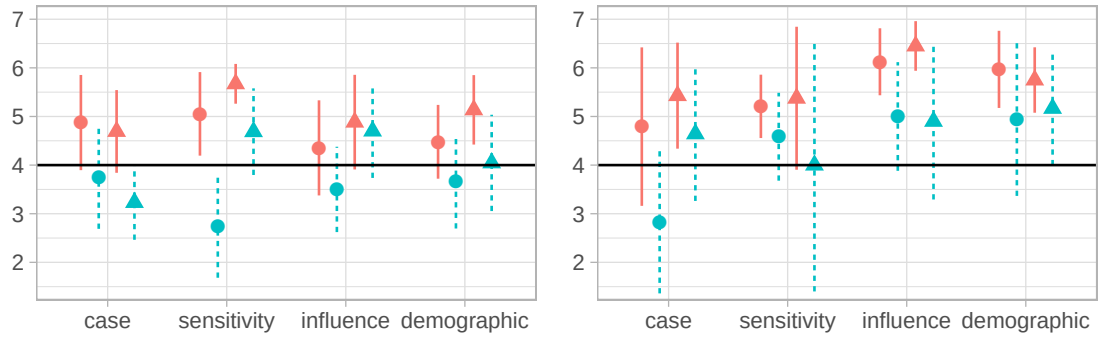


Figure 2.3: Same data as Figure 2.2 (Y-axis is fairness rating of *this* decision), split by prior position on the fairness of using the race feature. Left: Participants that consider using race "Unfair" *in general (prior_race_pos* < 4, 107 individuals). Right: Participants that consider using race "Fair" or neutral *in general (prior_race_pos* ≥ 4, 53 individuals).

2.6.2 Individual differences

We enter the following factors into the model: prior position on using machine learning to assist decision-making (*ML position*), prior position on fairness of using the race feature (*race position*), and *need for cognition*. We start from four level interactions of each of the individual difference factors with the three manipulated variables (explanation, data processing, disparate impact), and then iteratively reduce each factor to lower-level interactions if it is not significant. We eventually arrive at a model with the following terms: a four way interaction between race position and the three manipulated factors, $F(3, 144) = 2.59$, $p = 0.05$, and a marginally significant two-way interaction between ML position and explanation style, $F(3, 137) = 2.43$, $p = 0.07$. We did not find need for cognition to make a difference and removed it.

By including these individual difference factors in the model, we now find the three-way interaction between explanation style, data processing, and disparate impact to be significant, $F(3, 144) = 2.96$, $p = 0.03$ (its lower-level two-way interactions as well). In addition to the main effect of data processing ($F(1, 137) = 4.68$, $p = 0.03$) and disparate impact ($F(1, 144) = 28.86$, $p < 0.001$) as in the original model, we also find a main effect of ML position ($F(1, 137) = 17.31$, $p < 0.001$), race position ($F(1, 137) = 6.43$, $p = 0.01$), and a marginally significant main effect of explanation style, $F(3, 137) = 2.11$, $p = 0.10$.

The above significant three-way and four-way terms, after including race position in

the analysis, demonstrate that the consideration of this individual factor "de-noised" the data. In other words, it is only when an individual considers using race to be unfair, that a sensitivity-based explanation like this–"*If Nolan had been 'Caucasian', he would have been predicted to be NOT likely to re-offend*"–heightens the concern and significantly lowers the perceived fairness. When an individual does not consider it problematic to use race as a decision factor, they would not perceive such an explanation negatively. This trend is illustrated in Figure 2.3, where we separate participants who considered the race factor unfair and those who considered it fair-to-neutral (33.1% of all participants). In fact, for those who consider race to be a fair or neutral feature to use (Figure 2.3, Right), they did not perceive predictions made on the raw data (circles) to be less fair than processed data (triangles), and they generally rated fairness to be higher (thus the main effect of prior position on race).

The main effect of ML position and its interactive effect with explanation style indicates that a general positive position on algorithmic fairness enhanced perceived fairness, and also led to different explanation preferences. We conducted pairwise comparisons between styles of explanation, and found this interactive effect with ML position to be significant for influence-based explanation, $F(1, 148) = 6.25$, $p = 0.01$, and marginally significant for demographic-based explanation, $F(1, 148) = 2.77$, $p = 0.10$, if using case-based explanation as the reference. It is significant for influence-based explanation, $F(1, 148) = 3.73$, $p = 0.05$, if using case-based explanation as the reference. This implies that people who trust ML systems gain *even higher* confidence in the fairness of a prediction given global explanation (Figure 2.4).

It is worth noting that after controlling for these individual factors, we now see a marginally significant main effect of explanation style. Pairwise comparisons show that case-based explanation was rated marginally significantly less fair than influence-based ($F(1, 153) = 3.51$, $p = 0.06$) and demographic-based explanation ($F(1, 148) = 3.20$, $p = 0.08$). We consider it as evidence that case-based explanation is seen as generally less fair.

To summarize, in response to RQ1 and RQ2, we found evidence that: 1) Case-based explanation is seen as generally less fair; Global explanations further enhance perceived fairness for those who have general trust for machine learning systems to make fair decisions. 2) Local explanations are more effective than global explanations at exposing case-specific fairness issues, or fairness discrepancies between different cases. Sensitivity-

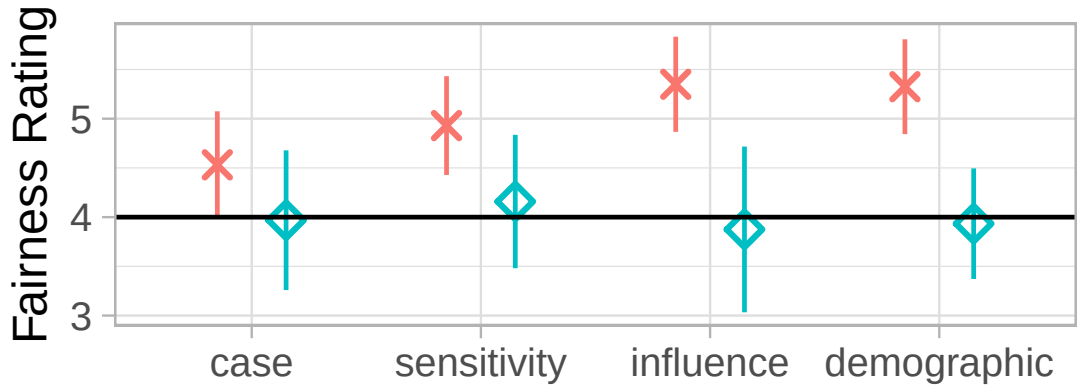


Figure 2.4: Overall mean fairness ratings, broken down by prior position on "Trust in ML" *(high trust=×, low trust=◊).*

based explanations are the most effective in exposing the fairness issue of disparate impact made by a particular feature—but only if the individual views using that feature as unfair. 3) In general, we show that individuals' prior position on ML trust and feature fairness have significant impact on how they react to explanations, and possibly more so than differences in cognitive styles.

2.7 Results: Qualitative

Along with collecting fairness ratings, we asked participants to justify their judgment. The authors reviewed this data and used open coding to extract themes in the answers. Here we discuss two groups of themes. One is to understand how participants made fairness judgments. Another is on participants' feedback for the four styles of explanations.

2.7.1 How is fairness judgment made?

In the open-ended answers, we investigated the criteria participants employed to judge fairness. We see variations in reliance on the provided explanations, and depth of reasoning about the algorithm's processes, providing further evidence of individual differences in the *criteria* used to make fairness judgments of ML systems.

2.7.1.1 General trust or distrust in ML systems

Some participants provided reasons not specific to a case or explanation, but that general trust or distrust of ML systems dominated their judgment, as they tended to give consistent ratings across cases. Reasons for a general trust include "*based on objective data is better than subjective opinions*" (CR-31)[7], "*large data set*" (CR-37), "*uses statistics based on prior knowledge to make a judgment*" (IR-176). In contrast, some participants considered generalization by statistics unfair–"*it might be unfair to group everybody together - makes more sense for the judge to have individual judgment.*" (IP-184), while others think that "*there needs to be a human element to the decision*" (CR-62). These observations corroborate Binns et al. [16] and further validate that participants varied on their general position on using ML system for criminal justice, and this influenced their fairness judgment.

2.7.1.2 Features used

Participants frequently cited features used by the algorithm as reasons for fairness or unfairness. Some explicitly differentiated between the process of the algorithm and the feature considered–"*The software makes decisions based on it's algorithm, so I believe it is fair and impartial on that account. However, some of the categories it is programmed to consider, such as age and race, are unfair*" (SP-71). It is interesting to note that we observe individual differences in the position on the fairness of race feature in the qualitative results as well. While many participants called out the problem of considering race, a few participants who saw the processed data commented that "*[if] race was not a predictor [it] may not accurately reflect the reality*" (DP-68). There is also some controversy on using age and juvenile priors as features. Participants' comments echo results from a previous study [58] showing that people consider multiple dimensions (e.g., relevance, disparate outcome, volitionality) in their judgments about the fairness of features used in decision-making algorithms, and individuals weigh these dimensions differently.

[7]Participant IDs give treatment info, explanation (**S**ensitivity, **C**ase, **I**nput-Influence, **D**emographic) followed by data processing (**R**aw, **P**rocessed).

2.7.1.3 Lacking features

As observed in [16], participants criticized the limited features used in our simple model. Some indicated interest in more detailed information on current features, such as "*frequency of priors or the interval of time since the last prior in order to get a more accurate assessment of what one's prior record means*" (DP-53). Others are less optimistic about the possible sufficiency of features to ensure fairness–"*software cannot fully take into account environmental factors that cause people to go down a bad path*" (CR-76).

2.7.1.4 Prediction process

Many participants based their fairness judgment on their understanding of the algorithms' process. Some, especially those presented with global explanations, closely examined explanation details, e.g. "*Software seems to be flawed in major areas... improper weighing of distant vs recent past, and a questionable choice of how to evaluate probabilities in each case*" (DR-119). Some also considered failure to account for external factors, e.g. "*the number may be relatively accurate for the race and charge degree categories, but if the [past] laws were different they would probably be higher*" (DR-107). Moreover, multiple participants attributed their low fairness ratings to insufficient understanding of process, or "*'how' the data is used*" (DR-172).

2.7.1.5 Data issue

A few participants questioned the underlying data used. Almost all of them were in either the demographic- or case-based explanation conditions, as these two styles leverage information about distributions of similar cases to explain the decision. For example, "*'Not re-offend' rate for African Americans is a little low. I think the percentage may be higher in reality... data could have been biased*" (DR-107).

2.7.2 Explanation styles

Below we summarize codes that are prominent for each explanation style. These results could help us better understand the benefits and drawbacks of each explanation style, and inform future work on designs of ML explanations.

2.7.2.1 Influence based

This is a global explanation that faithfully describes how each feature contributes to the algorithm's decision-making process. We observed that this explanation prompted comments on details of the process, such as the weights of different features, and the trends with regard to different categories of a feature, e.g. "*it is fair because it doesn't discriminate by race, but rather on age and prior convictions... if someone exhibits a behavior pattern it is likely to will continue, and I think people who are young are more apt to take risks*" (IP-208). On the one hand, detailed description of the algorithm process adds to the confidence in participants' judgments, which may help explain its enhancement of fairness perception among those trusting ML algorithms. On the other hand, it exposes more information to scrutiny, and it is thus subject to critiques from the heterogeneous standards of fairness.

2.7.2.2 Demographic based

This is a type of global explanation that does not expose the *process* of the algorithm, but justifies the decision based on the data distributions. Sometimes, the distributions were seen as convincing, e.g. "*The high percentage of people with more than 10 prior convictions who end up reoffending was staggering, and justifies the prediction*" (DP-57). Other times, participants found its explanation of the process inadequate, as the percentages do not clearly connect to an outcome–"*The percentages aren't high enough. It could go either way*" (DR-157). Sometimes it also directed participants' attention to the potential biases of underlying data.

2.7.2.3 Sensitivity based

The main benefit of sensitivity-based explanation seems to be conciseness and explicitness in directing attention to features relevant to the particular decision. It appears to be convincing and easy to process when a decision is uncontroversial –"*The rationale is so basic (no prior offenses) that it has to be fair*" (SR-138). "*It's taking into consideration everything that we would and puts it into an easy to read manner*" (SP-220). Consistent with our quantitative results, for disparately impacted cases where the race factor is explicitly mentioned, sensitivity-based explanation heightens the concern and

was perceived most negatively–"*It says that in the same situation, if the offender were African-American rather than Caucasian, they would have been likely to offend. This is racial profiling and inaccurate in my opinion.*" (SR-124).

2.7.2.4 Case based

As we found in the quantitative results, case-based explanation was judged to be the least fair—and the qualitative results provided reasons. First, some found it to provide little information about *how* the algorithm arrives at a conclusion. Second, the number of identical cases and the percentage of cases supporting the decision are often considered too small to justify the decision–"*It was unfair for the defendant because she was compared to only 22 other identical individuals... not to mention that only a little over 50% reoffended.*" (CR-61). This observation is consistent with Binns et al. [16], however, our work is based on the actual output of a ML model trained on a real dataset – allowing us to empirically show a limitation of case-based explanation[8]. Lastly, we found variations in individuals' positions on the fairness of the "explained process" (as opposed to the actual algorithm process) to make decisions based on identical cases. While some people consider it to be fair to "*compare the actions of people with similar history and backgrounds*" (CP-200), others questioned the underlying rationale such as "*is anyone really identical if more things considered*" (CP-201).

2.8 Discussion

2.8.1 Supporting the various needs of fairness judgment

The most important take-away from our study is that there are multiple aspects and heterogeneous standards in making fairness judgments, beyond evaluating *features*, as studied in previous work [58]. Our experiment highlights two types of fairness issues: unfair models (e.g., learned from biased data), and fairness discrepancy of different cases (e.g., in different regions of the feature space). Our qualitative results further illustrate

[8]We found that 16% of the test data exhibited the failure mode of *contradicting* the claim ($< 50\%$ of individuals with identical features share label). Meanwhile *insufficient justification* of the claim (between 45% and 55% label matches) was quite common, with 24% of the test data. The prevalence of these failure modes indicates inherent "unsoundness."

that algorithmic fairness is evaluated by various dimensions including data, features, process, statistical validity, as well as broader ethical and societal concerns.

Our results highlight the need to provide different styles of explanation tailored for exposing different fairness issues. For example, we show that local explanations are more effective in exposing fairness discrepancies between different cases, while global explanations seem to render more confidence in understanding the model and generally enhance the fairness perception. Hybridizing the two techniques reveals a possible human-in-the-loop workflow; using global explanations to understand and evaluate the model, and local explanations to scrutinize individual cases.

It is critical to note that different regions of feature space may have varied levels of fairness and different types of fairness issues. This calls for development of fine-grained sampling methods and explanation designs to better support fairness judgment of ML systems. To that end, we envision an active-learning paradigm for fairness improvement, where the system interactively queries the human for fairness judgment of its predictions, together with explanation options, then optimizes the algorithm based on feedback.

Our qualitative results suggest another useful categorization of explanation styles: *process oriented vs. data oriented explanation.* The case- and demographic-based explanations we studied leverage information on data distribution to justify their decisions but they reveal less about *how* the decision was made. Influence- and sensitivity-based explanations link each feature to the decision. We observe a general preference for process-oriented (*how*) explanations, although a focus on data has the potential benefit of directing attention to issues in the data and dilutes the "blame" on the algorithms.

2.8.2 Individual differences and descriptive fairness

Another contribution of our study is to empirically demonstrate how individuals' prior positions on algorithmic fairness impact their reaction to different explanations. We differentiate between a general position on algorithmic fairness and a position on fairness of a particular feature used.

The difference between normative (prescriptively defining what is fair) versus descriptive fairness and its implication for algorithmic fairness has been discussed in previous work [58]. Empirically, we show that even though race is considered a protected variable, individual positions on its fairness still vary (close to one third of participants considered

it neutral or fair to use). This indicates a lack of agreement on the meaning of moral concepts, a result Binns et al. [16] hinted at qualitatively. In different contexts, an algorithm developer may have to choose between a normative or a descriptive position of fairness, and it is important to be aware of the variation of fairness position in the population. For example, if a ML system takes a normative position and aims to eliminate pre-defined biases based on people's feedback, it may need to account for their prior positions to weigh the feedback differently. It may be arguable whether explanation should always attempt *soundness* and *completeness* for all individuals. On the other hand, if a system aims to provide optimal decision support for individual needs, it would be useful to provide mechanisms for individuals to express their prior positions as direct input for the algorithm (similar to the idea of active-learning by tuning features [145, 161]).

2.8.3 Limitations

We performed our study with crowdworkers, rather than judges who would be the actual users of this type of tool. Additionally, there are many styles and elements of explanations not studied here. One example is *confidence*, which we declined to present to participants because we could not control for it.

2.9 Conclusion

Our work provides empirical insights on how different styles of explanation impact people's fairness judgment of ML systems, particularly the differences between a global explanation describing the model and a local explanation justifying a particular decision. We highlight that there is no one-size-fits-all solution for effective explanation–it depends on the kinds of fairness issues and user profiles. Providing hybrid explanations, allowing both overview of the model and scrutiny of individual cases, may be necessary for accurate fairness judgment. Furthermore, we show that individuals' prior positions on algorithmic fairness influence how they react to different explanation types. The results call for a personalized approach to explaining ML systems. However, specific to fairness, ML systems may need to take a normative or descriptive position in different contexts, which may differentially require corrective or adaptive actions considering individual differences in their fairness positions.

Chapter 3: Addendum to "Explaining Models: An Empirical Study of How Explanations Impact Fairness Judgment"

Working on this paper was instrumental in my thinking about XAI. In particular, we wrote several workshop papers discussing the experience further, examining both the role of templates [44] and the arguments against the case-based explanation style we had deployed [41]. Here, we will paraphrase the arguments from those two papers and make a few other observations.

First, to conclude the "Case Against Case-based Explanation" paper [41], I argue that explanations should incorporate the use of the decision boundary, holding up the sensitivity-based explanation as a more promising contrast. Consistent with this point, during my thesis defense, a committee member demonstrated that this goes a bit deeper, past what we had covered in Section 2.8.1 about the difference between explaining the *data* vs explaining the *process* (which in our case is enacted by the decision boundary).

In particular, the demographic-based explanation suffers from a lot of the same flaws as the case-based explanation described in the "Case Against..." paper because it does not use the decision boundary. To illustrate the importance of the semantic difference between explaining decision processes and explaining data, consider the following: Suppose the raw classifier is the same logistic regression model as we used in the paper, but the *processed* "classifier" is simply a function that returns "Likely to reoffend" for all inputs. With those two different processes (different decision boundaries), the case-based explanation would be exactly the same, because the data are exactly the same. This would also be true of the demographic explanation. The reason is that these explanations do not take into account how the process (decision boundary) relates to the example to be explained.

Delving deeper into case-based explanations, the argumentation from the "Case Against..." paper [41] begins by defining terms: I consider a case-based explanation to simply retrieve training example(s) based on nearest neighbors from the example being explained. Then, I argue against using this style of explanation using four points. First, I do a short survey of literature, and mostly finding that the case-based explana-

tion style seems to have a lot of negative empirical results. Second, I show that distance in some embedding space may not actually mean much, due to both geometry and weak causal linkage. Third, I argue that case-based explanation offers weak evidence when it is trying to support a strong claim, since it is essentially a proof by example. Fourth, I describe how in some settings, privacy or sheer data set size may restrict the presentation of training examples to the user.

Second, in the "Templates" paper [44], we started by observing that the process of surveying human participants had been labor intensive for the participants. Then, we considered that people often deploy AI in domains with very large state/action spaces. As a result, it will possibly be infeasible to have humans scrutinize each possible decision, so "Satisfaction[1]" studies as described by Hoffman et al. [69] will be inherently limiting in such domains. The alternative, according to the same paper, is evaluating the "Goodness[2]" of an explanation.

One mechanism that assists Goodness evaluation is the *template* (Figure 3.1 shows an example), which sets the form of the explanation before system fills the individual values for a decision. We made a key cross-cutting realization from working on this paper and considering other projects: in our internal research meetings, we were mostly discussing templates—not explanation instances. Thus, when devising and formalizing Goodness mechanisms, basing them on templates and not instances would likely become important later.

Third, at the time we wrote this paper we had not drawn a clean conceptual separation between what "has the property?" and what "is viewing the property?" (the property is the aspect of the system the explanation is intended to reveal, here it was fairness). To concisely state where I eventually landed on these two questions: the decision process "has" the property, and the explanation and task combine as the lenses used to "view" the property, meaning they may add their own influence to what the user sees. Thinking about this more led to the "resolution" approach described later in Chapter 6, and thinking more after that work has led me to believe that it will eventually be important to conceptually separate *task* resolution from *explanation* resolution.

[1] I like to think of *Satisfaction* metrics as akin to asking an *empirical* question, like "does it help somebody do a task?".

[2] I like to think of *Goodness* metrics as akin to asking an *analytical* question, like "does somebody *think* it might help somebody else do a task?".

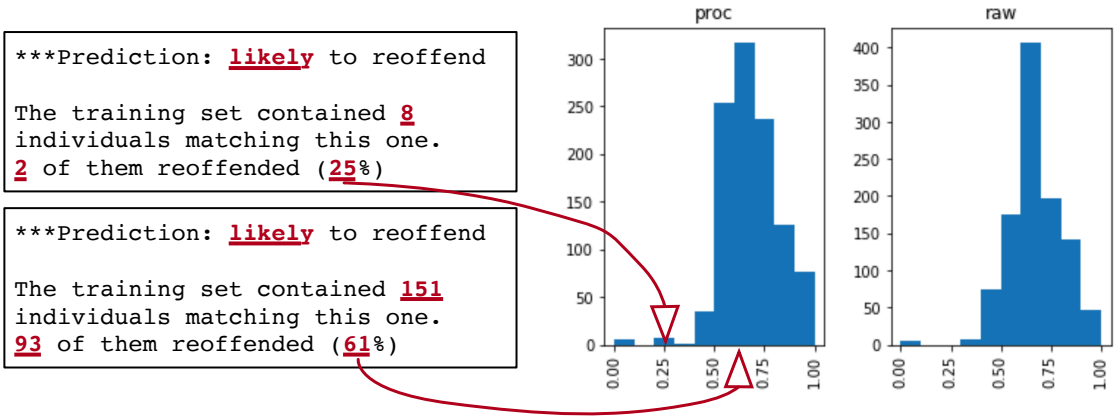


Figure 3.1: Figure from [44]. *Left:* Two example explanations provided by the system used by Dodge et al. [46]. The black text is static and the added red highlights show text that will be based on calculations about the input—intended to show how explanation templates get filled in. It demonstrates how the Hoffman's Satisfaction [69] results vary based on the input (e.g. the top explanation is far less convincing). *Right:* Histogram of the matching percentages underlined in Figure 3.1, for the classifiers trained on raw and processed data. These histograms show how differently the two classifiers behaved, but also show an interesting result—namely how often case-based explanation self-refutes (by providing low %s), or does not substantiate any claim (by giving near 50%, in a binary classification setting). However, this insight might not have been observable for a user under Hoffman's Satisfaction [69] formulation, as users typically only work with a small number of explanations and self-refuting ones are rare.

Chapter 4: After-Action Review for AI (AAR/AI)

By: 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

Appeared in: ACM Trans. Interact. Intell. Syst. 11, 3–4, Article 29 (December 2021), 35 pages.
DOI: https://doi.org/10.1145/3453173

Abstract: Explainable AI (XAI) is growing in importance as AI pervades modern society, but few have studied how XAI can directly support people trying to *assess* an AI agent. Without a rigorous process, people may approach assessment in ad hoc ways—leading to the possibility of wide variations in assessment of the same agent due only to variations in their processes. AAR, or After-Action Review, is a method some military organizations use to assess human agents, and it has been validated in many domains. Drawing upon this strategy, we derived an AAR for AI, to organize ways people assess reinforcement learning (RL) agents in a sequential decision-making environment. We then investigated what AAR/AI brought to human assessors in two qualitative studies[1]. The first investigated AAR/AI to gather formative information, and the second built upon the results, and also varied the type of explanation (model-free vs. model-based) used in the AAR/AI process. Among the results were (1) participants reporting that AAR/AI helped to *organize their thoughts* and *think logically* about the agent; (2) AAR/AI encouraged participants to reason about the agent from a *wide range of perspectives*; and (3) participants were able to leverage AAR/AI with the model-based explanations to *falsify* the agent's predictions.

[1]This paper is a revised and expanded version of [115].

4.1 Introduction

By design, AI systems perform decision-making on behalf of a human user. This means that in safety-critical applications such as self-driving cars, vendors may take on additional liability when things go wrong. Failures may have such grave consequences that they are likely to wind up in court [31]. Was the accident caused by the driver not reacting in time, or a defective AI? [125]. How can AI stakeholders best determine that an AI system is safe and regulation compliant?

Given that intelligent agents interact with the world in ways analogous to those of human agents, could established techniques for evaluating the quality of human performance be applied to an AI? In this paper, we investigate this approach by adapting a technique called After-Action Review (AAR) for use with AI. AAR was devised by the U.S. Army in the mid-70's [121], and has been a success in various branches of the military. It has also been adapted for other domains including medical treatments [157], transportation services [111], and fire-fighting [76]. Apparently, most of these adaptations have proven successful, as a recent meta-analysis of 61 studies reporting effect size for use of AAR found a moderate practical effect overall [85].

We term our adaptation AAR/AI (pronounced "arf-eye", short for "AAR for AI"). AAR/AI is a *process* for *application domain experts* to use in assessing whether and under which circumstances to rely upon an AI agent. We envision AAR/AI to be suitable for sequential domains, guiding the human through a series of steps to evaluate an AI agent's actions using explanations.

To investigate AAR/AI in the hands of human users, we set it in the context of a real-time strategy (RTS) game as the sequential system. We created a custom game in StarCraft II (Section 4.4.1). Then, we created a reinforcement learning (RL) agent that yielded high-quality actions in the domain (Section 4.4.2). For this agent, we also devised two types of explanations of the agent's actions (Section 4.3.4)—one type was a Model-Free explanation and the other was a Model-Based explanation—so as to observe AAR/AI with two types of agents.

Model-Free and Model-Based agents work differently, yielding different possibilities for explanation. Model-Free agents simply compute a value for each considered action and then select the maximum. In contrast, Model-Based agents expand a search tree as they perform action selection. The root of the tree reflects the current state of the

system. The agent considers many actions by predicting the state transition each action will cause. The process is then repeated using each predicted state as a starting state. Model-Based agents offer a richer space for explanation because the action and state information available in the tree is human-interpretable.

To improve experimental control, we needed Model-Free and Model-Based explanations which select the same actions. To accomplish this, we used the same Model-Based agent for both, so that they encoded the same policy, then heavily pruned the Model-Based explanation tree to form the Model-Free one, only exposing the information that an Model-Free system would have.

AlphaZero [165] is a classic example of a Model-Based system. It uses MCTS to expand a game search tree—the agent uses its model of the game rules to recursively predict subsequent states as part of the decision process (given the current state and potential agent actions as input). Model-Free agents too can be applied in domains which are sequential, but are more commonly deployed in domains which are not, such as image classification (e.g. VGG-19, illustrated by [65]'s Figure 3).

To investigate AAR/AI in the context of these explanations, we conducted two qualitative studies. Study One employed a one-on-one in-lab think-aloud design with paper prototypes and focused primarily on the process. Using our Study One results, we implemented an interactive prototype and ran Study Two. Study Two allowed us to both triangulate[2] with our preliminary results and to consider AAR/AI with two types of explanations: a Model-Free explanation and a Model-Based explanation.

With our two studies, we investigated the following research questions:

RQ1 (Study One) To what extent are participants able to make sense of and learn from our explanations while using AAR/AI for assessment?

RQ2 (Studies One and Two) Which actions should be included in search tree explanations? How do these design choices affect user interaction patterns?

RQ3 (Studies One and Two) How did the aspects of theory present in our explanations affect participants' ability to make explanation-informed statements?

RQ4 (Study Two) How did differences in how participants engaged with our explanations affect their cognitive load?

[2] *"In the social sciences, triangulation refers to the application and combination of several research methods in the study of the same phenomenon."* Source: `https://en.wikipedia.org/wiki/Triangulation_(social_science)`

4.2 Background & Related Work

There are many papers describing the challenges of evaluating AI systems' quality (e.g. [26, 57]), including specific attacks (e.g. [49]). Rising to meet these challenges, approaches like DeepTest [179] attempt to apply software engineering concepts to improve testing of deep neural networks. In particular, they seek to measure and improve "neuron coverage" (proposed by Pei et al. [134], similar to code coverage). To accomplish this, they apply a series of input transformations, a form of data augmentation conceptually similar to fuzzing. However, these approaches are *system*-oriented in terms of exposing problems, not *human*-oriented by giving an assessor the tools to determine appropriate use.

4.2.1 People Analyzing AI

Human-oriented evaluation of AI is an active area of research, though much of it is at a different granularity than we needed. For example, Lim et al. researched how their participants sought information in context-aware systems powered by decision trees. The result of their research was a code set of several "intelligibility types" describing the information. They discovered that their participants demanded *Why* and *Why Not* information, especially when the system behaved unexpectedly [107]. Using Lim's code set, Penney et al. studied how experienced RTS players looked for information when understanding and evaluating an "AI," but they found that participants preferred *What* information over *Why* information and that the large action space of StarCraft II led to high navigation costs, which meant missing important game events [136]. Dodge et al. analyzed how shoutcasters (human expert explainers, like sports commentators) assessed competitive StarCraft II players. They showed the ways that shoutcasters present information that they thought their human audiences needed [47]. Kim et al. gathered 20 experienced StarCraft II players to play against competition bots and rank them based on performance criteria. They noted how human evaluations of the AI bots differ from the evaluations used for AI competitions and that the human player's ability plays a huge role in their evaluations of the AI's overall performance and human-likeness [90]. These studies found how people evaluate an AI, but they did not present a *structured process* for assessment.

There are several models which consider system assessment in a human-oriented way; however, these works do not provide an assessment process for AI, but rather on whether humans will *adopt* systems or not. One such framework is Technology Acceptance Modeling (TAM) [37]. TAM can predict how well a system will be accepted by a user group and explain differences between individuals or subgroups. More recently, the UTAUT (Unified Theory of Acceptance and Use of Technology) model was proposed as an acceptance evaluation model [66]. Carrying on this spirit, recently researchers have produced a spate of publications based on need-finding or perception interviews meant to identify barriers to adoption (e.g. [16, 202, 71, 23]). While these techniques can assist in assessment, they do not offer a concrete *process* for human assessors to enact.

More recently, a few researchers (e.g. [9, 195]) have made first forays into guidelines that can be used to assess explanations without the user-in-context[3] required by adoption models. Yang et al., identified two main challenges that may explain the rarity of this kind of fundamental AI usability research, *"uncertainty surrounding AI's capabilities... [and]... AI's output complexity."* [204]. As with adoption models, guidelines can assist in explanation assessment, but do not offer a concrete process to follow.

4.2.2 People Explaining AI

The primary purpose of explanations is their ability to improve the mental models of the AI systems' users. Mental models are *"internal representations that people build based on their experiences in the real world"* that assist users in predicting system behavior [124].

Devising explanations that actually lead to better mental models is an active area of research. One such example is Kulesza et al.'s proposed principles for explaining (in a "white box" way) machine learning (ML) based systems, wherein the system made its predictions more transparent to the user [97], which in turn improved the quality of their participants' mental models. Another study by Anderson et al. [11, 13] provided insights into the variability of changes in the mental models of participants with different explanation strategies of an AI agent. One promising explanation strategy is to manage users' expectations. For example, Kocielnik et al. found that interventions, like adding

[3]Hoffman et al. [69] describe two kinds of AI evaluation, based on "Satisfaction" (roughly speaking, 'does it help a user complete a task?'), and "Goodness" (roughly speaking, 'does a panel of experts think it is good?').

a gauge, helped participants estimate the system's accuracy [94].

Another direct consequence of altering the mental models of users is the improvement in their ability to control the system. According to a study by Kulesza et al. [99], participants with the most improved mental models were able to customize the system's recommendations best. Roy et al. found that participants preferred high controllability, even in low accuracy settings [153]. Wang et al. set up an accuracy-control tradeoff explicitly in their auto ML system, allowing users to search longer for higher accuracy, or adjust the search constraints for higher control [196]. Another kind of tradeoff, posed by Smith-Renner et al., reports that the systems *adhering* to the user input more often can increase *instability* with respect to other changes that occur when the model updates to incorporate that input [168], a problem also reported by Stumpf et al. [172]. Still, the preference for controllable systems seems to hold even when the controls *do not work* [186].

Explanations in the domain of AI agents in RTS games have been gaining traction in recent years. In a study by Metoyer et al. [117], they present a format where experienced gamers played while providing explanations to non-RTS players, finding that one key to the explanation process was the manner in which expert players communicated while demonstrating how to play. The study by Kim et al. [91] had experienced gamers play against AI bots in order to assess the bot's skill levels and overall performance. However, despite the research mentioned above, there is a dearth of literature concerning what humans really *need* in order to understand and assess such systems [131].

4.2.3 After-Action Review

To structure our assessment method, we turned to processes that have been used for humans to assess *other humans*, including Post-Control, Post-Project Appraisal and After-Action Review (AAR) [159]. Our criteria for the process to use as our basis included: (1) have a structured and logical flow, (2) be well established, and (3) be suitable for evaluation *during* a task, not just useful at the end of a task. We selected the AAR method as the one that best fulfilled these criteria.

AAR is a debriefing method created by the United States Army, and it has been used by military and civilian organizations for decades [156] to encourage objectivity [111]. The purpose is to understand what happened in a situation and provide feedback, so

US Army AAR Process
Introduction and rules.
Review of training objectives.
Commander's mission and intent (what was supposed to happen).
Opposing force commander's mission and intent (when appropriate).
Relevant doctrine and tactics, techniques, and procedures (TTPs).
Summary of recent events (what happened).
Discussion of key issues (why it happened and how to improve).
Discussion of optional issues.
Discussion of force protection issues (discussed throughout).
Closing comments (summary).

Table 4.1: Steps of the US Army AAR process [185].

people can meet or exceed their performance standards by going through the structured series of steps shown in Table 4.1.

The AAR process was primarily used as a method to provide performance feedback after soldier training sessions. Before starting an evaluation session, the leader (a designated individual across all sessions) performs groundwork to collect and aggregate data from the training session for further analysis. The leader enters that session with a pre-planned mechanism to collect data. The evaluation session begins by reiterating the objectives of the analyzed exercise. From there, the leader asks a series of open-ended and leading questions about what happened during the training session, making sure to encourage a diverse range of perspectives. These responses are then filtered into a recapitulation that the group collectively agrees on, and the discussion is shifted to any shortcomings in performance. This is followed by brainstorming solutions to avoid or improve responses to problematic outcomes. The evaluation concludes by delineating an action plan to adhere to for future training [185].

AAR showed effectiveness for combat training centers [156], and the military still uses it, with a recent investigation of current methodologies for simulation-based training [62]. Outside military applications, AAR has been used in other domains, from medical treatment [142, 157], emergency preparedness [36], and emergency response [76, 102]. The closest research to ours discusses how AAR will be different for manned-unmanned team compositions, but focused on the technologies needed to support the AAR process, not the process itself [19].

4.3 The AAR/AI Process

Our After-Action Review for AI (AAR/AI) is an assessment method for a human assessor to judge an AI. We base the steps of our method from Sawyer et al's DEBRIEF adaptation from the Army's AAR [157]. In their adaptation, they Define rules, Explain objectives, Benchmark performance, Review what was supposed to happen, Identify what happened, Examine why, and Formalize learning. Table 4.2 outlines our AAR/AI adaptation.

The original AAR method is a facilitated, team-based approach, but our AAR/AI method is for an individual that is reviewing, learning the AI's behavior, and assessing its suitability [159]. The outcomes are different for the approaches: AAR aims for transfer of knowledge within a team, and AAR/AI aims for individual acquisition of knowledge and assessment of an AI. These two primary differences between AAR and AAR/AI are what generated the specific ways AAR/AI (Table 4.2's columns 2 and 3) carries out the original method's steps (Table 4.2's column 1).

4.3.1 AAR/AI: Defining Rules & Objectives

A facilitator starts each session with a tutorial on the user interface, domain, explanations, and the objectives of the assessment (Steps 1-2, Table 4.2). This contextualizes the discussion in terms of what the assessor is supposed to do and the objectives of the agent that they are assessing. After that, the facilitator begins the AAR/AI "inner loop" (discussed next), and after every loop is done the assessor completes a questionnaire.

4.3.2 AAR/AI's Inner-Loop: What, Why, How

During each iteration of the inner loop, the facilitator asks the assessor, "*What was supposed to happen?*", "*What happened?*", "*Why did it happen?*", and "*How can it be improved?*" (Steps 3-6, Table 4.2). The assessor also summarizes what happened in the past three rounds and writes down anything they observed that was good, bad, or interesting on an index card. At Step 5, we provided the assessor with the AI's explanation for the most recent round, and asked them to explain why the AI did the things it did, according to the process in Table 4.2. Next, to formalize learning about this particular decision, the facilitator asks the assessor the questions listed in Table 4.2

	AAR Steps	AAR/AI ?s Answered	AAR/AI Empirical Context
	1. Define the rules	How are we going to do this evaluation? What are the details regarding the situation?	We established the rules of evaluation and the domain (see Supplemental Materials).
	2. Explain the agent's objectives	What is the AI's objective or objectives for this situation?	We explained the AI's objectives for the situation (see Supplemental Materials).
AAR/AI Inner Loop	3. Review what was supposed to happen	What did the evaluator intend to happen?	We asked, *"What do you think should happen in the next round(s)?"*.
	4. Identify what happened	What actually happened?	The participant watched the required number of rounds. Then, we asked, *"Could you briefly explain what actually happened in these past rounds?"*.
	5. Examine why it happened	Why did things happen the way they did?	We asked, *"Why do you think the rounds happened the way they did?"*. Next, the participant summarized anything good, bad, or interesting on an index card. Last, we provided the participant the agent's explanation.
	6. Formalize learning (end inner loop)	Would the evaluator allow the AI to make these decisions on their behalf? What changes would they make in the decisions made by the AI to improve it?	We asked three questions: *"Would you allow the AI to make these decisions on your behalf?"*, *"What changes would you make in the decisions made by the AI to improve it?"*, *"Would you allow the Friendly AI to make this category of decisions on your behalf?"*.
	7. Formalize learning	What went well, what did not go well, and what could be done differently next time?	The participant completed a post-task questionnaire (see Supplemental Materials).

Table 4.2: How AAR/AI (right two columns) adapts the original After-Action Review debrief steps (left column). The "Empirical Context" column explains how we realized it in Study One. (Study Two's realization of AAR/AI was almost the same, except that we shortened Step 6 to just the "What Changes" question.) Note that steps 3-6 form an "inner loop" that we repeated every three decisions. The parts outside the inner loop are documented in our Supplemental Materials (tutorials, questionnaires, etc), so we describe them only briefly.

step 6, (e.g. whether they would allow the AI to make these decisions on their behalf). Thus ends the inner loop, which would repeat until the end of that analysis session.

4.3.3 AAR/AI's Artifacts

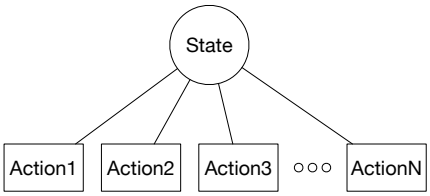
Part of AAR/AI involves creating materials to help keep everyone on task during the assessment. The US Army AAR uses cards in order to log observations [185], though the information collected is largely focused on personnel and their positioning. Since the AI performs within the RTS domain, we turned to how professional shoutcasters analyze AI, like AlphaStar [170]. They used formatted text for actions that they found "good," "bad," or "interesting," which we replicated in the AAR/AI's index cards. This prevents assessors, regardless of the AI's use, from relying on memorizing when a decision is good or not. By using such written artifacts, the AAR/AI process has the benefit of gaining retrospective feedback on the process itself or the explanations used in it. Further, artifacts like these can assist in comparing the assessment results from multiple individuals or be released with the system as a means to document the kind of validation conducted and the results from it, akin to Model Cards [119].

4.3.4 AAR/AI: Explanation Component

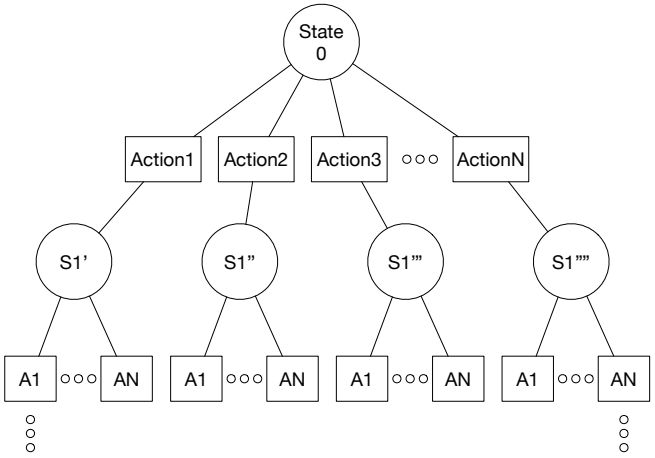
AAR/AI evaluators, like the AAR equivalent, require information on what happened, so our process requires an embedded Explanations Component, since the evaluators not only must they know *what* happened, but the agent must be able to explain *why* it performed an action. In both studies, we used a model-based agent to enable a model-based explanation.

A model-based agent (and its explanation) offers the benefit of explicitly representing its emerging model of the world, such as the future states the agent is trying to reach or avoid—in essence, an underlying rationale for its decisions[4]. For example, consider Model-Based agents that expand a search tree as they select actions, allowing them to fill in an explanation template [43] like the one shown in Figure 4.1b. On the other hand, Model-Free agents can only fill in a more limited explanation template, illustrated

[4]To compensate for model-free agents' lack of underlying rationale, one body of research attempts to generate approximations of an underlying rationale, e.g., [48].



(a) Template for a Model-Free explanation, notionally showing its "search stump," because it has no depth—just values associated with taking an action in a state. Here, since we are showing the template, we have simplified away details like state/action representation and presentation (addressed in Appendix B).



(b) Template for a Model-Based explanation, notionally showing the full search tree. In general, these search trees do not reach terminal nodes, and to evaluate the quality of non-terminal nodes, the agent uses a heuristic function or Model-Free system.

Figure 4.1: A comparison of explanation templates which can be filled out by Model-Free and Model-Based explanations, respectively. Note that if one imagines starting with just a root node and the best action, and iteratively revealing the tree via interaction, Model-Free explanations would only need to "widen" the tree, while Model-Based explanations also support "deepening". We will return to this in Section 4.8.1.

in Figure 4.1a. In particular, Model-Free explanations do not expand a search tree—instead more of a search stump—by attaching to each action only a single number for its value.

We therefore prototyped a Model-Based explanation for Study One, capturing a portion of the agent's search tree. That explanation is shown in Appendix B, and here we focus on the revised version of our Model-Based explanation, used in Study Two, shown in Figure 4.2. We described the search tree to participants as, *"...a diagram of decisions, where the Friendly AI decides what actions or decisions it must take to complete a round in the game."*

The explanation lays out the agent's "explanatory theory" [167] of how the game could play out in different situations. In essence, the theory's "constructs" of that theory are: game states, roles (e.g. friends or enemies), actions available to various roles, and

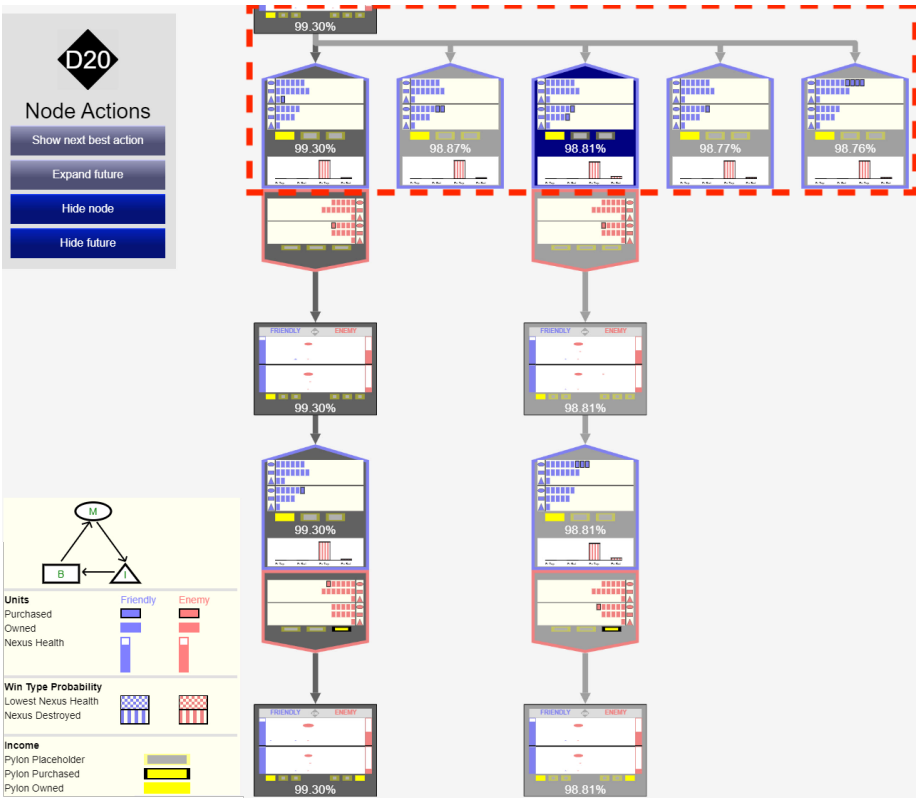


Figure 4.2: Interactive Model-Based explanation for DP 20 in Study Two, as observed by participant S1MB20. The Model-Based explanations, shown above, starts at the top with the current state node. Next are the top five predicted friendly agent actions considered, each followed by the enemy agent action that is predicted to be most effective. Next down are the predicted states that are consequent to those actions. The cycle is then repeated again. We refer to a fully-rendered prediction trajectory as a "future". The principal variation, the best predicted future, is at left, with actions decreasing in estimated value to the right. Participants could choose to adjust visibility of individual nodes or future trajectories of particular actions by selecting nodes and clicking buttons on the Node Actions menu. The legend at bottom left reminded participants of the meaning of each rendered state and action detail, as well as showing the rock-paper-scissors relationship among unit types. We simulated Model-Free explanations (dashed red box) by featuring only the root node and the friendly agent action nodes directly below, essentially truncating the depth of the tree. Even though participants in both treatments were able to increase or adjust the *width* of the tree, the Model-Free explanations were essentially different in that they provided no information beneath the friendly agent action node. Since Model-Based explanations included the tree at greater depth, they allowed participants to expand the width of the tree at any internal node—as opposed to just the top level actions.

(estimated) values of different states and actions.

In AAR/AI then, the evaluator's central mission is to evaluate one aspect of the AI agent's theory: its falsifiability [140]. To carry out this mission, the evaluator answers the AAR/AI questions (e.g., What just happened? Why? ...) by gathering information from a combination of game behavior and the explanation's diagram of actions and states among which the agent is deliberating (Figure 4.1). To falsify the agent's theory in some way means that the evaluator has discovered a flaw in the agent's theory (reasoning).

4.4 Methodology Shared by Study One and Study Two

To inform our design of AAR/AI, we ran two in-lab studies: Study One, a one-on-one think-aloud qualitative study and Study Two, a two-treatment qualitative study run in small groups. The main goal of Study One was to formatively investigate participants' sensemaking attempts when doing AI assessment and how AAR/AI came together with those attempts. Additionally, since the AAR/AI process embeds an explanation, we designed both studies to include investigating the explanation strategy in the context of the process.

Study Two moved beyond the sensemaking goal of Study One to gain insights into how the explanations might help humans with *failure detection* or *fault localization*. To illustrate, Study One featured *only* the Model-Based explanation strategy in paper prototype form, whereas Study Two's treatments included software implementations of both the Model-Based explanation and the (simulated) Model-Free (detailed later in Section 4.4.2). Notionally, since Model-Free agents do not expand a search tree with any depth and can be thought of as a search stump, Model-Free explanations are limited to fewer interactions than Model-Based (as illustrated in Figure 4.1).

Both studies used the same domain and agent implementation, which we describe next.

4.4.1 The Domain

StarCraft II is a popular Real-Time Strategy (RTS) game that offers hooks for AI development ([190, 192]) and a flexible engine for map creation[5]. Using this engine, we built

[5]Many map creation resources are available at places such as [177].

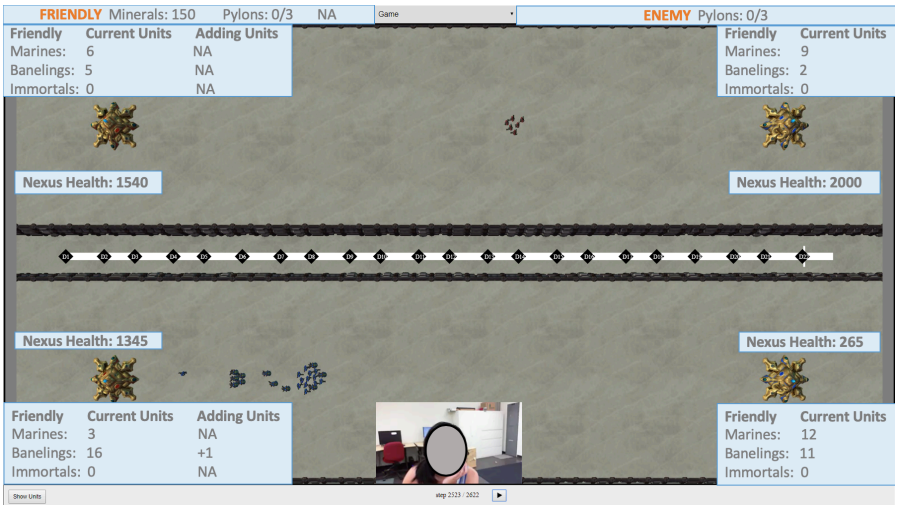


Figure 4.3: Game screen at decision point 22 during Study One. Note the text boxes offering state information (current units, nexus health, etc) as well as action information (adding units). The evaluation interface primarily adds a time slider (shown in the middle of the screen with a diamond for each DP) and the blue overlays to increase visibility of fonts presenting information available in the in-game interface.

a custom game called Tug of War, shown in Figure 4.3. The objective of the game is to destroy either of the opponent's Nexus in the top lane or bottom lane. If no Nexus is destroyed after 40 rounds (or decision points, which we denote as DPs), the player whose Nexus has the lowest health will lose.

At every DP of the game:

- Each player receives income (100 minerals, +75 per pylon)
- The player chooses to build any combination of unit production facilities (i.e. barracks) to be added for the next DP, subject to the following constraints:
  1. Total cost cannot exceed current mineral count
  2. Players are only allowed to build in *one* lane at a time
  3. Players do not know the opponent's action until both actions are finalized
- Players spawn units equal to the total number of unit production facilities currently held (i.e., 5 barracks $\rightarrow$ 5 marines)

At each DP, both players choose which lane to build in and the number of unit-producing buildings to spend resources on for each of 3 unit types which share a rock-

paper-scissors relationship. **Marines** (50 minerals) are low health units that attack in small quick shots. They are effective against immortals. **Banelings** (75 minerals) are medium health units that attack by exploding on contact. Banelings are effective against marines. Lastly, **Immortals** (200 minerals) are high health units that attack in large slow shots. Immortals can inflict significant damage on a Nexus. Players may also choose to build a pylon to increase their income. The maximum number of pylons they can build is 3, and the cost of a pylon increases each time one is purchased. Note that an action in this context is essentially an integer vector representing the intended purchase of unit-producing buildings and/or pylons, meaning the branching factor is combinatorial with respect to minerals possessed.

Once units spawn, the players can no longer control them; they will move toward the enemy Nexus and attack any enemies along the way. Also, units *always* spawn at the same location each wave.

4.4.2 The Agent Implementation

We have pointed out that the agent used for both studies is Model-Based, meaning it has access to a transition function that maps a state-action tuple to the successive state. Applying the transition function allows the agent to expand a search tree, and perform minimax search on it[6]. The system uses three learned components (all represented by neural networks): the transition model, the heuristic evaluation performed at leaf nodes, and the action ranking at the top level.

The heuristic evaluation function estimates the value, or quality, of non-terminal leaf nodes in the search tree. This function is necessary to address the depth of the full game tree, since the search will rarely be able to expand the tree to the point where all leaf nodes are terminals. The action ranking function provides a fast estimate of the value associated with taking each action in a state. This function is necessary to address the large action-branching factor by only performing the more expensive tree expansion under some number of top-ranked actions to improve estimates (similar to AlphaGo and AlphaZero [164, 165]). A big difference, however, is that our system uses a learned transition model, due to the stochastic and complex nature of the transitions between states; whereas Silver et al.'s used a perfect move-transition model (e.g., Chess's

[6]For more information on game tree search, see Russell and Norvig, Chapter 5 [155].

deterministic rules).

We actually used the same Model-Based agent in both treatments of Study One and Study Two by simulating the Model-Free treatment's agent for the Model-Free treatment. We simulated the Model-Free explanation simply by withholding the Model-Based agent's learned "model of the world," which amounted to less completeness, excluding information past the value associated with the actions (illustrated in Figures 4.1 and 4.2). This design enabled scientific control, with the Model-Free and Model-Based agent choosing from (and selecting) the same actions given the same state. This level of control would have been extremely difficult to ensure via independent training processes.

4.5 Methodology Specific to Study One

For Study One, we recruited 11 students at Oregon State University who had not taken classes in AI/ML or participated in our previous studies. Since our game is based on StarCraft II, we recruited those familiar with real-time strategy games, to ensure that participants could understand the game sufficiently to assess the AI.

A researcher served as facilitator with one participant (assessor) during the AAR/AI process, starting with a tutorial on the interface, domain, and task (covering AAR/AI Steps 1/2). Since each session was limited to 2 hours, we wanted to ensure that each participant reached the end of the task with time for our post-task questionnaire. Thus, we decided to have them analyze every third decision point of the 22 available, including the last one (e.g. 3,6,...,21,22). This allowed up to 5-7 minutes for each iteration of the AAR/AI inner loop—though it was rarely necessary to enforce limits during the studies. We chose to sample these decisions because timing our pilot participants revealed that we would not have time to cover all of them, and we wanted the participants to see the full evolution of a game.

At each iteration, the researcher asked the assessor a structured series of open-ended questions to elicit their thoughts as they performed their assessment of the AI's actions (Steps 3-6). Additionally, the participant wrote on index cards (Section 4.3.3) to help them formalize thoughts and offer the option to refer back to previous notes.

Upon completion of the task (Step 7), we asked: *"Did the process of the questions I asked you help you understand and assess the AI better?"*, *"Do you think the AI's diagrams have enough detail?"*, *"Would you prefer the width of the diagram to be narrower*

or wider? Or do you like the way it is?", *"What kind of actions would you have liked to see on the diagram?"*, and *"In the main task, did you find these cards useful?"*. Finally, we compensated participants $20.

Each session spent ∼30 minutes for the briefing/tutorial (pre-task), ∼50 minutes on the inner-loop (the main-task), and ∼25 minutes on the post-task questionnaire. This timing was consistent with Sawyer et al.'s recommendations (25/50/25%, respectively) [157].

4.5.1 Analysis Methods

To answer **RQ1**, we drew from a code set that Dodge et al. used in their StarCraft II study, which had been adapted from Lim et al.'s work [108, 47]. Dodge et al. also added a "judgment" code, which the AAR/AI needed because of the nature of assessment. Individually, the two researchers coded 20% of the data corpus, achieving an inter-rater reliability (IRR) of 76.4%, computed via Jaccard Index [77]. Given this level of reliability, they then split up the remaining coding.

To answer **RQ2** and **RQ3**, two researchers applied content analysis [72] to the coded statements from the post-task questions about helpful or problematic elements of the process or explanations, resulting in the code set in Appendix A, Table A.1. The two researchers coded 21% of the data corpus separately, achieving inter-rater reliability (IRR) of 82.4% (Jaccard). Given this level of reliability, they then split up the remaining coding.

We enumerate these code sets in the context of the relevant results sections.

4.6 Results: Study One

We begin with participants' sensemaking attempts when they were using AAR/AI and our explanations, deferring Study One results that intertwine with Study Two results to Section 4.8.

4.6.1 Using AAR/AI to Learn

The goal of our project was to enable participants to understand how the AI agent is "thinking" well enough to evaluate how suitable the agent is for different situations that arise—which involves people building mental models of the AI agent. In this subsection, we consider what the AAR/AI process brought to our Study One participants' mental-model building.

In a post-study questionnaire, we directly asked Study One participants what was helpful about AAR/AI and what was not. In their responses, many of the Study One participants commented on how AAR/AI's "structuredness" helped their understanding by keeping their thinking organized, structured, and/or logical. (Only a single Study One participant said it was not helpful, but this was because they believed that with their experience in RTS games, they already understood the AI's behavior without the need of any assistance.) For example:

> S1MB8: *"Uh, yes, I would say <AAR/AI was helpful>. It definitely directed me towards what I should be paying attention to."*
>
> S1MB18: *"I could think what it should improve on and why the previous round happened the way it did. So, when those questions were broken down... Really helped in following the game."*
>
> S1MB14: *"...it categorized the flow of logic that we should've had in analyzing the prediction and what actually happened, so it kept it more organized, and therefore, more logical."*
>
> S1MB17: *"I know it was too much information ... it helped me understand it better. ...it just helps me ... to understand it better, and makes it more logical."*

To understand the level of our Study One participants' mastery of understanding the agent, we applied Bloom's Taxonomy [17], which is a framework used by educators to categorize the different levels of learning. The taxonomy has six levels [14], ranging from basic understanding of a concept (level 1), through a fairly advanced understanding (level 6). Each level requires learners to engage with a higher level of abstraction than the last. The application of Bloom's taxonomy to our context is detailed in Table 4.3.

Level: [17]'s Description	**How it applies to understanding the AI**	**Examples from our participants**
1. **Remembering**: Have students acquired the ability to correctly recall information?	Participants *recall* domain information, such as game rule(s), what an agent can do with particular game units, etc. (Supported by AAR/AI's questions about the game.)	+S1MB20: *"It'd probably buy another baneling... to counter the marines..."*
2. **Understanding**: Can students understand information they have learned to recall?	Participants *understand* the domain information provided. (Supported by AAR/AI's "What" and first "Why" question.)	+S1MB8: *"...you <the AI> don't necessarily know which lane they're coming through... it's not much of an informed decision until the first round happens."*
3. **Applying**: Can students apply their newly learned knowledge?	Participants *apply* the explanation of the AI to the game. (Supported by second "Why" question.)	+S1MB2: *"I ...like it how <the explanation diagram> is, because like I could try to draw my own conclusions from it rather than just like 'oh this is just what happened'."*
4. **Analyzing**: Can students see patterns and make inferences about a problem?	Participants *analyze* the AI's problems in the game, and reason about solving them. (Supported by the prediction task and the "What changes would you make" question.)	+S1MB2: *"So the bottom one did pretty well like overpowering the enemy AI and even attacking nexus, lowering its health while the top one, the enemy AI did a better job sending more marines and the friendly AI sent banelings which got overpowered by the marines."*

		+S1MB19: *"So we have almost same health on top and bottom. So, to defeat us, they have to focus on either one. So I guess they will focus bottom, because they have to save them at the time. I guess we have to use minerals to buy immortal here, so that we can save ourselves and at the same time, kill the enemy."*
5. **Evaluating**: Can students take a stand or decision, and justify it?	Participants *evaluate* the AI agent, and judge if they would allow the agent to make decisions on their behalf in this or similar situations. (Supported by the "Would you allow..." question series.)	+S1MB5: *"Producing these banelings <in both> lanes allowed nexus damage bottom lane, and then having the one or two marines do consistent damage on the nexus really took down the nexus health, so that was actually a really good decision."* +S1MB20: *"This is gonna be sad. Yep. It's all downhill from here. (after watching the replay) Uh, the friendly AI lost, uh, due to their misinvestment in the top row, and only increasing their baneling count, which only works at melee range which is ineffective to marines if there's already a baneling wall in front of them."*
6. **Creating**: Can students create a new point of view?	Participants *create* new points of view by generalizing upon, abstracting above, or recommending differences in the AI's behaviors.	+S1MB14: *"Well, the enemies will invest in banelings, and I feel that the friendly's will invest in marines, especially more in the top row, since it is more damage..."* +S1MB21: *"I would consistently save a small quantity of minerals each round, rather than trying to save them all in a single round."*

Table 4.3: Bloom's taxonomy levels Study One participants achieved in learning the agent's behavior.

As Table 4.3 shows, subsets of Study One participants showed mastery of every Bloom's level. In fact, each of these participants achieved Bloom's Level 5 at least once during the study. Further, all except one of them achieved Bloom's Level 6 at some point.

Bloom's Level 5 is of particular interest to our project: it is the level of understanding that allows evaluation. Evaluation is precisely the level of understanding needed for assessing an AI.

In considering how the participants who reached Bloom's Level 5 managed to do so, we turned to the Lim-Dey intelligibility types, which we used as a codeset for our qualitative coding (Table 4.4). As the results show, each of AAR/AI steps guided participants' thinking (according to their self-reports) toward different Lim/Dey perspectives [107]. The wording of the AAR/AI questions compared with the Lim/Dey type names may explain some of this result.

For example, the AAR/AI question in the top row of Table 4.4, "What ...should happen," guided most participants to focus on "What Could Happen"—an almost syntactic match between the AAR/AI question and that Lim/Dey type. The "Why...did" AAR/AI question (fourth row) also featured a strong syntactic match with the Lim/Dey "Why did" type. While not a near-syntax match, the AAR/AI question on the last row, "What changes would you make...to improve it," is still semantically a reasonable match to the "How To" Lim/Dey type.

The AAR/AI question on the second row, "what ... actually happened," is more subtle. This question guided many participants to focus on Output types of information. In the context of a computer system, this still seems a fairly direct semantic match between the question and Lim/Dey type. However, this question also guided over one-fourth of the responses toward the Input type, which has neither a syntactic nor semantic match to the Lim/Dey type. It could be an example of these participants working through a cause/effect connection.

Other research has shown each intelligibility type has its own advantages and disadvantages (e.g. [108, 34]), so we see the diversity of perspectives that AAR/AI seemed to elicit as a particular strength of AAR/AI.

	What	What Could	How To	Judgment	Why Did	Why Didn't	Inputs	Model	Outputs	sum
"What do you think should happen in the next 3 rounds?" (Before watching them)	2	71	16	1	0	0	24	6	2	122
"Could you briefly explain about what actually happened in these past three rounds?" (After watching them)	13	6	2	6	18	2	53	12	74	186
"Why do you think the the rounds happened the way they did?"	2	6	3	1	32	2	24	31	30	131
"Why do you think the Friendly AI did what it did?" (After seeing the explanation)	2	8	8	0	55	1	60	27	36	197
"What changes would you make in the decisions made by the Friendly AI to improve it?"	3	8	56	2	2	0	38	3	2	114
Sum	22	99	85	10	107	5	199	79	144	750

Table 4.4: Lim Dey coding of participant responses, sliced by question asked during the AAR/AI.

4.6.2 Participants' Views of Model-Based Explanations

Participants' mental-model building with AAR/AI relied upon the presence of an explanation. In Study One, the model-based tree diagrams provided participants with a global view of the agent's decision process, supplementing the local-only "right now" view provided by the game state. As two participants put it:

S1MB2: *"I kinda of like it how it <explanation diagram> is, because like I could try to draw my own conclusions from it rather than just like 'oh this is just what happened'."*

S1MB14: *"<In the game state>... difficult to grasp the whole situation, so having the graph gave me a chance to get my footing on overall trends and options."*

This way of using the explanation was a theme which was echoed in a post-task response from another participant:

S1MB17: *"The diagrams used to make it easier also helped to understand the predictions. To look at one thing from many angles and make appropriate pre-*

dictions."

However, a pitfall some participants fell into was extrapolating too much information from the tree diagrams. Several participants seemed *certain* about the agent's long-term plan, which was troubling because the explanation did not make such a plan explicit, if the agent even had one.

> S1MB21: *"At this point, I feel certain that the friendly's trying to destroy the bottom nexus of the enemy."*
>
> S1MB10: *"I think it's because it was a whole game plan from the beginning. ... like from the beginning of the bottom lane, the friendly AI started increasing the troop numbers."*

However, the explanation could not possibly have shown a many-step game plan, because the agent was only looking head two states.

Another participant also expressed difficulty in seeing long term strategies, but for a different reason—granularity mismatches between moves, tactics, and strategies:

> S1MB20: *"There are subtasks and decisions that go into making a strategy and not being able to see this had me make less informed assumptions about the future decisions."*

In Study Two, we built interactive software, in part to alleviate the problem of too much or the wrong information at the wrong time.

4.7 Methodology Specific to Study Two

Study Two used a similar protocol as Study One, but in lab sessions with up to 5 participants at a time and without the think-aloud protocol, to allow more participants than are viable with think-aloud studies. Study Two also utilized interactive software that we built, standing upon the results we had just learned from Study One. Study Two's prototype being implemented in software enabled participants to perform actions such as expanding the tree.

We recruited 22 participants for Study Two at Oregon State University using the same criteria as before, and randomly assigned them to our two treatments, Model-Free and Model-Based. Each participant made predictions, viewed the replay, then viewed the associated explanation for seven decision points (DPs 6, 7, 11, 17, 20, 26, and 36), selected due to their having sizeable impacts on the game outcome, which is the friendly AI winning the game at DP 37. We gave participants four minutes to fill out

the prediction sheet, two minutes to understand the explanation for each DP, and an additional four to complete the questionnaire with questions from Table 4.2's Step 5. To ensure that they did not advance to the explanation before we were ready, we had participants type a short unlock code into the interface after the researcher provided it verbally.

4.7.1 Analysis Methods

We analyzed **RQ1, RQ2, and RQ3** using the same codesets described in Section 4.5.1. For **RQ4**, two researchers looked for evidence of participants having been somehow informed by the explanations, in the written responses to our AAR/AI questions (What happened, what was Good/Bad/Interesting about it, Why did it happen, and What changes would you make). Specifically, we removed responses without clear evidence that participants had been informed by the *explanation*, as opposed to the game state or a participant's domain knowledge. Each part of good/bad/interesting was a separate response, so the 22 participants answered 6 questions each for a total of 132 responses, of which 50 passed the filter to be considered "Explanation-Informed Statements". Using content analysis [72], we then derived the code set shown in Table 4.6. A different two researchers coded 44% of the data corpus independently, achieving IRR of 81.2% (Jaccard). (Usually, researchers use a smaller subset of the data for agreement, but we expanded beyond the more typical 20% in order to include more instances of rare codes.)

4.8 Results: Both Studies

Since Study Two was intended to complement and triangulate with Study One, we present Study Two's results in combination with the pertinent results from Study One[7].

4.8.1 Which Information to Show?

To answer the AAR/AI questions, participants needed information from the explanations, but which information and how much of it to show is a question XAI researchers

[7]Keeping context explicit is the reason we prefix each participant ID with the appropriate study number; e.g., S1MB5 denotes "Study One, Model-Based participant 5".

have been wrestling with for years (e.g., [47, 108, 107, 99, 97, 195, 128]). One participant simply wanted to see everything—corresponding to an explanation with maximum completeness:

S1MB5: *"All the possible actions and all possible outcomes."*

Unfortunately, with the agent considering combinatorial action spaces, showing the full search tree all at once (statically at least, it might be possible to navigate via dynamic mechanisms) would have been too large for humans to process. Thus, we needed to choose a smaller set of noteworthy actions to show—but which ones and how many?

Recall that the explanations showed only four actions (Figure B.1). Some participants thought there should be more and/or different actions. For example:

S1MB5: *"... since there are only four options ... if it was a possibility for more options 'cause there was definitely more possibilities."*

However, these four options were only "top" according to the agent's estimations, which may not have been the right four:

S1MB5: *"I would think the AI would have the best four, which it didn't have the best four."*

One participant proposed also showing the *worst* possible choice:

S1MB20: *"I'd like to see ... what the friendly AI thinks is the ... choice that would give them the least chance of winning as well as their greatest chance of winning..."*

Study Two participants seemed to need information about another class of action as well—actions that spend *all* resources—since not explaining this class led them to believe the AI did not consider these actions carefully enough:

S2MF46: *"Why didn't AI use all remaining resources?"*

S2MF38: *"It's unreasonable to not purchase buildings when you've got no reason to save and invest in pylons."*

S2MB30: *"There is no reason that I can think of for it to have not spent minerals."*

Despite the fact that this class of actions was in these participants' world view, the AI does not include this human-created abstraction in its world view. That said, the agent *does* consider each available action, so the "complete" search tree contains at least some information about the kind of actions the participants describe—even if they were pruned away. The importance of this class of actions to the participants suggests that participants need this information, but answering this question might require finding more than just *one* action from the class, but instead *many* of them to reason about as

MF-PID	Widen	Drag
S2MF1	39	
S2MF32	15	
S2MF40	6	7
S2MF42	54	8
S2MF43	41	2
Totals	155	17

MB-PID	Widen	Deepen	Drag
S2MB20	14	12	
S2MB21		10	2
S2MB31	46	20	
S2MB35	1	2	
S2MB36	7	6	
Totals	68	50	2

Table 4.5: Interaction totals from participants who interacted with the explanation by: "widening" the tree by adding an action node (at any location), "dragging" a node in the tree by shifting its position, presumably to better enable comparison, or "deepening" the tree by expanding the future associated with a top-level action (refer to Figure 4.2). (Participant S2MB8's data was damaged, and thus excluded from this analysis.) The following 10 participants did not interact with the explanation beyond pan and zoom operations: S2MF38, S2MF41, S2MF44, S2MF45, S2MF46, S2MB23, S2MB26, S2MB28, S2MB30, S2MB39.

a set. This suggests that participants might benefit from query systems built to select *all* instances of a class of action interesting to their world view.

As to how many actions to show, seven of the participants indicated that they liked the tree—but one wanted a smaller one, and three wanted a larger tree.

S1MB8: *"I liked the way it is. It's easy to read."*

S1MB21: *"I do not have any problem with narrow diagram..."*

S1MB11: *"I would just have more options available..."*

The previous paragraphs discussed participants' self-reported responses. Now we turn to what Study Two participants *actually did* when provided with an interface, enabled by watching the screen capture videos from 21 of the participants[8].

Even given the interactive explanation, 10 of the participants (5 in each treatment) did not interact with the explanation beyond panning and zooming—in effect treating it as a static diagram too large for the screen. The behaviors of the remaining participants are aggregated in Table 4.5. Thus, the experiences of the 10 pan/zoom-only Study Two participants with the interactive explanations prototype were similar to what Study One's participants experienced with their paper-prototyped explanations. This suggests the importance of the system's initial/default presentation of explanations—for about

[8]One participant's data (S2MB8) was not included in this analysis, due to corruption of their video file.

half of our participants, our choice of initial presentation was the only one they ever looked at.

Of the Model-Free participants who did tree manipulations, most were to widen the tree more, which was one of the few interactions available. Tree widening usually occurred in one or two short bursts of 3–8 node additions to the tree. However, they also dragged more nodes around than the Model-Based participants did, presumably for the purpose of comparing actions. For example, at one point, S2MF40 performed a series of drag operations to visually group similar action nodes (characterized by a top lane action making 3–5 marines and 1–2 banelings).

Model-Based participants also manipulated tree width, but they also took advantage of the Model-Based capability of going deeper into the tree, to peer into the AI's predictions of the future. Expanding depth adds 5 nodes, but expanding width adds just 1 node, so the amount of additional information Model-Based participants added per "deepen" manipulation was 5 times as much as with a "widen" manipulation, so the participants who used "deepen" processed a great deal more information than those who did not.

Of these "deepen"s, the most popular among the participants was the one that expanded the second-best action, then the third-best, and so on (17 second-best, 14 third-best, 8 fourth-best, 3 fifth-best, 8 beyond fifth-best). (Recall from Figure 4.2 that the actions were ordered best to 5th-best.) One pattern shared by four participants (S2MB20, S2MB21, S2MB31, and S2MB36) was to expand the top k futures, for some k, then visually scan it up and down. This behavior "filled the screen" with information, suggesting that our explanation's default presentation did not adequately fill up its rectangular viewing region with nodes (it started out as roughly a $\vdash$ shape). Had we done so in this system, the "static diagram" participants might have passively consumed more, and the "screen fillers" would not have had to manually fill the screen.

4.8.2 Explanations as Theory

One way to think about how participants worked with the explanations to answer AAR/AI questions, is to view the explanations as the agent's "theories". In the explanation trees, upon reaching leaf nodes, the agent used a neural network to evaluate the quality of states. These estimates were, in essence, *axioms* and the minimax search

that proceeds atop those values are akin to *theorems*. Thus, if the axioms hold true, then the theorems were true.

In Study One, we saw that not all participants were willing to "grant the axioms." Some were:

> S1MB14: *"I mean because, those are the ones with greater scores. So I guess that is why it chose those decisions."*

Others did not grant them and found themselves not understanding or possibly disbelieving parts of the diagram.

> S1MB10: *"I think diagram needs improvement, because those are not that clear at some times. ...It does have enough details, but the decisions were, not made... according to the diagram."*

Two participants identified the issue well: that the win probabilities have no clear provenance.

> S1MB8: *"... If there's any easy way to say why it came up with these numbers... there were several steps that I just didn't know why it was taking that action..."*

We found that RTS experience seemed to be a potential driver for rejecting the heuristic evaluation function, with S1MB5 and S1MB20 being particularly critical of the agent's decisions:

> S1MB5: *"Wow, rewards went down... A baneling is better than a marine by rewards points, but there's clearly a better answer."*

Those with less RTS experience seemed less critical of the agent's explanation, but they still compared the agent's actions to the tree:

> S1MB14: *"Information didn't always line up with what occurred. Therefore, it gives a false belief on what/how the AI is doing."*

When we conducted Study Two, through use of the Model-Free and Model-Based explanations, we offered two very different presentations of the agent's theory. In particular, the Model-Free explanations are mostly leaf nodes, meaning they are almost entirely *axiomatic*. Despite this, some of Study Two's participants did find the Model-Free explanations helpful:

> S2MF43: *"Decision tree helped understand logic of AI better."*

Further, they were able to use Model-Free explanations to compare different actions, for example:

> S2MF41: *"It was very helpful to be able to see multiple potential game paths side by side."*

However, the Model-Free participants did not have access to the information that

would allow them to "disprove" deeply nested theorems by following them all the way down the tree. Recall from Figure 4.2 that Study Two's Model-Free explanations provide the current state at the root node, and then the top k actions and their values beneath that. In contrast, Model-Based explanations allow explorations all the way down the tree, eventually running into an axiomatic value, where we see the same curiosity about provenance that we saw in Study One:

S2MB30: *" Where does the % come from?"*

Thus, Model-Free explanations lacked some information that the Model-Based participants appeared to value highly:

S2MB21: *"Ability to see additional buildings for the next round gave insight on future AI actions. Explanation elements were easy to read and understand."*

S2MF41: *" I'm not 100% sure the information given in the explanations necessarily completely reflected the AI's decisions."*

One way of considering the value Model-Free vs. Model-Based participants obtained from the explanations is to consider their Explanation-Informed Statements. These are defined in Table 4.6 and, as the table indicates, Model-Free participants made fewer Explanation-Informed Statements of every type than Model-Based participants did (20 vs. 48). Further, Model-Free participants not only provided fewer of them than Model-Based participants did, but also did not even attempt Explanation-Informed bug reports until near the end of the task, as Figure 4.4 illustrates.

The bug reports were also different. Below are all four reports from Model-Free participants:

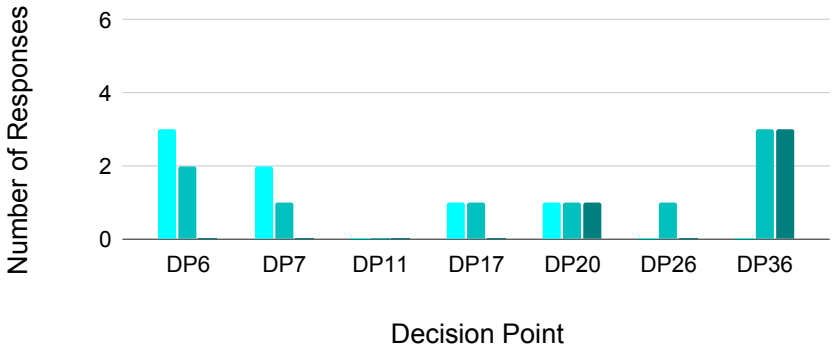
S2MF46: *"Good choice, but in bottom nexus is much lower. Why not commit to destroying it?"*

S2MF46: *"Why didn't AI use all remaining resources <at round 36>?"*

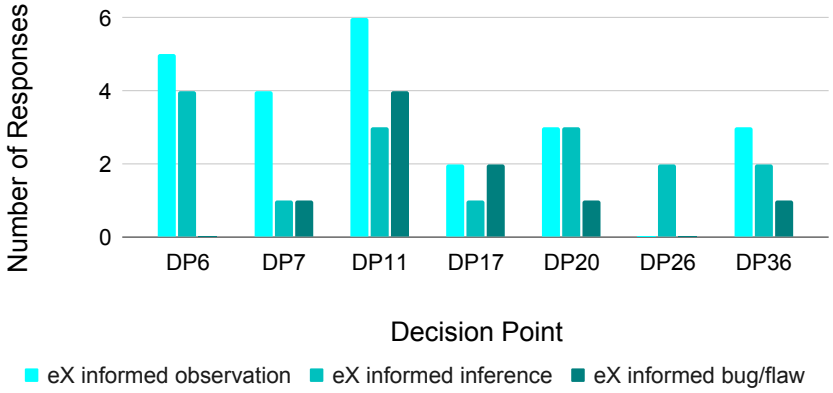
S2MF37: *"no round 36 purchase? Why?"*

S2MF38: *"It's unreasonable to not purchase buildings when you've got no reason to save and invest in pylons <at round 36 of 40>. I guess there is a bias introduced on how many buildings it can buy at a time."*

In essence, the Model-Free bug reports above are simply disagreements with high-level strategic choices the agent makes, as opposed to *falsifications* of the logic contained in individual nodes or transitions. A few Model-Based participants also gave those kinds of bug reports, such as:



(a) Explanation-Informed Statements from Study Two participants using **Model-Free** explanations.



(b) Explanation-Informed Statements from Study Two participants using **Model-Based** explanations.

Figure 4.4: Explanation-Informed Statements from Study Two participants interacting with Model-Free (top) and Model-Based (bottom) explanations. Statements are broken down into 3 categories. DPs (each bar cluster) are time ordered and aligned.

Code: Description	Example	MF	MB
Explanation-Informed Observation: Participant strictly interprets the explanation.	S2MB39: *"The friendly AI bought 1 baneling predicting that the opponent marines would increase."*	7	23
Explanation-Informed Inference: Participant forms or adjusts their mental model explanation, judge the explanation.	S2MB23: *"The AI is thinking ahead of how to win the game in the shortest number of rounds."*	9	16
Explanation-Informed Bug Report: Participant identifies flaw/bug in agent's reasoning from explanation OR finds explanation confusing.	S2MB26: *"Only 2 immortals were created. Prediction of marines was wrong."*	4	9
	Totals	20	48

Table 4.6: Explanation-Informed Statements code set, as applied to Study Two's 22 participant responses to our decision questionnaire (What happened, what was Good/Bad/Interesting about it, Why did it happen, and what Changes would you make), with examples and counts from each treatment.

S2MB30: *"There is no reason that I can think of for it to have not spent minerals."*

However, in addition to these strategy disagreements, Model-Based participants considered the correctness of individual predictions that go into the overall action selection:

S2MB26: *"Only 2 immortals were created. Prediction of marines was wrong."*
S2MB28: *"I think friendly AI is not able to assess that bottom lane is better. It is doing very well in bottom lane. But end result predicted is wrong."*

Or, stated just as logically but less passionately:

S2MB36: *"The friendly AI decided to fortify the bottom lane assuming an attack. The attack actually came from the top, where the enemy now has the advantage."*
S2MB28: *"The enemy AI outsmarted friendly AI. It sent marines along with banelings. Friendly AI thought enemy AI will send marines so it bought baneling producing building."*
S2MB36: *"It's interesting that the AI keeps assuming an attack on the bottom and not defense on the top."*

S2MB36: *"It still assumes it will win by destroying the top nexus."*

S2MB28: *"Friendly AI still predicts it will win by destroying top enemy nexus."*

These differences in the Model-Free vs. Model-Based participants' Explanation-Informed Statements illustrate a key strength of Model-Based explanations. They enabled Model-Based participants to "disprove" aspects of the agent's theorems by seeing inconsistencies and logic errors in the path propagating the *axiomatic* values computed at the leaves to the *theorems* about action selection. This process brings explicit falsification [140] capabilities to the system's users.

We observed participants engaging in falsification in both studies. In particular, Model-Based explanations make part of the search tree explicit and include concrete predictions about the future, including states. These concrete predictions allowed participants to falsify those predictions:

S1MB14: *"So the friendly had ... two banelings, so one baneling and some marines. Yes, that seems right. ... it predicted that the enemy would buy two more marines, and it ended up being so. Yep, it was right ... it was predicted that they would buy a baneling, and they did ... so far, it's going as predicted."*

S2MB26: *" Only 2 immortals were created. Prediction of marines <from the previous state> was wrong...<later>...Prediction was correct."*

We explicitly crafted parts of the process to allow the human to reflect on *their* past thoughts, but this participant focused on the accuracy of the *agent's* predictions about the future. Notably, this type of assessment was made possible by the Model-Based agent, and our explanations revealed relevant information to be able compare different time slices.

Thus far, we have focused on viewing explanations as theory in terms of their composition and falsification of elements. However, there are other criteria that can be used to evaluate theories [167]. In Table 4.7, we consider how to apply these criteria to evaluate *this* agent's Model-Based explanation, this *style* of Model-Based explanations, and in some ways, even *all* Model-Based explanations.

	"The degree to which..." [167]	**Applicable to...**	**Evidence to date for or against**
Testability	...empirical refutation is *possible*: constructs and <predictions> are understandable, internally consistent, free of ambiguity	...this *explanation* of the agent's model of the world.	*Empirical*: The agent's explanations were found to be understandable by several participants, as described in Section 4.6.1. The diagrams were clear and explicit in their information from most, but not all, participants' reports.
Falsifiability /Empirical Support	...is supported by empirical studies that confirm its validity	...this *explanation* of the agent's model of the world.	*Empirical*: Our explanations explicitly represented the agent's predictions about likely future states and their values, which participants could falsify.
		...this *style* of Model-Based explanation.	*Empirical*: AAR/AI evaluators (one instance: our participants).
Explanatory Power	...accounts for and predicts all known observations within its scope	...this *explanation* of the agent's model of the world.	*Empirical*: One measure is whether the agent's theory and explanation correctly predicted everything. In our study, the agent did not achieve this. *Criteria-based*: Whether its constructs are sufficient to express every possible action and state, i.e. completeness. In this study, the constructs have full explanatory power—but our explanation limited the number, so the actual explanation was not complete.
		...this *style* of Model-Based explanation.	
		...*all* Model-Based explanations.	

Parsimony	...<has> a minimum of concepts and propositions	...this *explanation* of the agent's model of the world.	*Criteria-based*: This explanation had 4 constructs/concepts that do not overlap, so cannot be reduced further.
Generality	...breadth of scope... and independent of specific settings	...this *explanation* of the agent's model of the world.	*Criteria-based*: This explanation's scope is limited to explaining this particular domain.
		...this *style* of Model-Based explanation.	*Criteria-based*: The style of explanation is not restricted to games, and should be usable for any sequential setting of Model-Based AI.
		...*all* Model-Based explanations.	Model-Based explanations are restricted to Model-Based agents.
Utility	...supports the relevant areas	...this *explanation* of the agent's model of the world.	*Empirical*: Most, but not all, participants reported the agent's explanations to be useful to understanding its actions.
		...this *style* of Model-Based explanation.	*Empirical*: AAR/AI evaluators (one instance: our participants).

Table 4.7: Applying Sjøberg et al.'s Evaluation Criteria for Theories [167] to the agent's Model-Based explanation

4.8.3 Participants' Cognitive Load and Performance

How did the differences in how participants engaged with the different explanations play out in participants' views of the challenge, effort and frustration levels of the entire AAR/AI process they experienced? To provide insights into this question, we turn to the NASA Task-Load indeX (TLX) responses from 15 of the Study Two participants (not all participants provided this data) at the end of the session. The TLX is a validated

	Mental	Physical	Temporal	Performance	Effort	Frustration
Model-Free	14.5	1	13	13.5	15.5	3
Model-Based	11	3	13	14	13	9

Table 4.8: Median results of the NASA TLX. Our discussion focuses on the responses with the greatest differences between the two treatments (highlighted): Mental Demand, Effort, and Frustration.

post-task survey to measure cognitive load [63]. As shown in Table 4.8, participants rated Physical Demand very low. There was also no difference in Temporal Demand (in either the medians or the distribution of data points) and little difference in Performance. Thus, we ignore those three and shift focus to the remaining factors.

The remaining three factors reflect participants' perceptions of cognitive load. Table 4.8 suggests that the participants who saw the Model-Free treatment tended to feel more Mental Demand[9] than the Model-Based participants in their AAR/AI-based evaluations. Consistent with this result, Model-Free participants also reported higher Effort[10] than the Model-Based participants. However, the Model-Free participants reported *less* Frustration[11]. This observation was unexpected for us, since Model-Free explanations contain a smaller amount of information.

These three results conceptually relate to Sweller's influential cognitive load theory [175]. Cognitive load theory includes three concepts: *intrinsic load*, i.e., the cognitive work that is inherent in the task for everyone; *germane load*, i.e., helpful additional cognitive work that may be necessary for that individual (e.g., inferring helpful new abstractions, such as by comparing a past item with a current item to abstract above the current situation); and *extraneous load*, i.e., extra, *un*helpful cognitive work that hampers the individual in performing the task (e.g., having to continually look up the meaning of different UI widgets) [175, 132].

Using these concepts, Mental Demand ("task-inherent" load) approximates intrinsic load, and Effort ("your" load) approximates the sum[12] of intrinsic + germane + extrane-

[9]TLX question: *"How mentally demanding was the **task**?"* (emphasis added)

[10]TLX question: *"How hard did you have to **work** to accomplish your level of performance?"* (emphasis added)

[11]TLX question: *"How insecure, discouraged, irritated, stressed, and annoyed were you?"*

[12]Orru et al. discussed a version of the NASA-TLX modified to equate the Effort question specifically with extraneous load [54, 132]. However, without that modification, NASA-TLX's Effort question is not confined to extraneous load

ous load [132]. Our results suggest that some participants decided that Mental Demand matched Effort (i.e., I had to do it, so it must have been what the task needed). Frustration is an interesting side-effect relating to Demand and Effort—our data suggested that it reflected participants' reaction to excessive load, especially extraneous load.

4.9 Discussion

4.9.1 Future AAR/AI Adaptations

AAR/AI is highly adaptable, and this provides leeway to iteratively improve it. Two areas for improvement that we observed were that participants thought they could remember what happened in the past, and that participants found questions/artifacts repetitive and burdensome at times. For example:

S1MB20: *"... I am fairly confident in my ability to remember what occurred."*
S1MB5: *"Some of this stuff kind of repeats..."*

An alternative might be to instead enable people to decide where to pause, in an approach similar to the empirical mechanism used by Penney et al. [136]. In that study, their participants watched a replay until they came to a decision that seemed important, at which point they could pause, consider our questions, and write down their thoughts. In essence, blending this device with our inner loop would give more control to the evaluators as to how often and exactly where the evaluation questions need to be answered.

As a meta-analysis of AAR by Keiser and Arthur [85] observed, it *"was initially operationalized with high administrative and content structure..."* with the goal being that *"higher administrative structure is expected to free up cognitive resources that would otherwise be spent on how to conduct the AAR."* Further, the authors go on to describe situations with less structure, and offer a flowchart (see Figure 10 from [85]) to help select the appropriate flavor of AAR for a variety of use cases.

Our short series of studies left many open questions about AAR/AI's efficacy in different possible usages. Among them, to what extent is it: *...rigorous* enough to support examining catastrophic failures that will necessarily consume hours of time from investigators? *...efficient* enough for real time analysis, akin to sports commentary? By investigating open questions like these, researchers will be able to discover shortcomings

and devise adaptations to improve fitness for different usages—and possibly illuminate other evaluation processes in so doing.

4.9.2 Prediction as Explanation

Trend 1: People used explanations as prediction tools. Reed et al. suggested that explaining a solution to a problem helps people solve similar problems [146]. Our strategy followed a similar approach, where participants predicted the agent's action (i.e., the problem), saw the action (i.e., the solution), and then provided an explanation to the action (i.e., explanation of the solution). Some participants even began using the explanations as the basis for their prediction:

> S1MB8: *"Understanding the diagram gave some insight into how the AI thought, which made predicting its next move easier."*

Participants engaging with the Model-Based explanation reported attitudes consistent with a series of studies Kelleher and Hnin observed, *"suggest that learners who attempt to understand the steps of a problem solution may have higher germane load but improved ability to apply these elements in novel situations."* [87].

Trend 2: The process of having participants predict the actions first, and then showing them the actions, was powerful. Another trend we observed was that predicting the AI agent's decisions *prior* to observing the AI agent's actual actions turned out to be part of our *explanation strategy*. One of the pillars of learning effectively is self-explaining [29]. Those researchers describe how students who learn with understanding the material and forming self-explanations on their own achieve better outcomes than those relying heavily on examples to learn and struggling to generate explanations on their own. Positioning the prediction task before the observation task effectively caused participants to create self-explanations for the AI agent's actions. Participants used the process and the explanation, to generate their own explanation for predicting the agent's actions:

> S1MB10: *"I think the aim of the AI is to increase the number of minerals, and then go to the last one that is immortals, so that they can make a great damage to the nexus."*

Participants who answered AAR/AI questions perform a "rationale generation" [48] task, which appears to offer some benefits as an AI evaluation strategy.

Renkl et al. found that acquisition of transferable knowledge can be supported by eliciting self-explanations [148]. Learners with low levels of prior topic knowledge profit from such an elicitation procedure. We observed this effect in our study, as participants with little experience in RTS comfortably navigated through the process of assessing the AI's actions—even forming their own explanations.

4.9.3 Encouraging Metacognition

Researchers in the field of education have long pointed to the benefits of metacognition, in which learners evaluate the success of their own learning/understanding processes [51]. Metacognitive activity is well-established as an important influence on learning and understanding [197].

Participants in Study One and Study Two, with both the Model-Free and the Model-Based explanations, showed several instances of metacognition that seemed to come from the integration of AAR/AI, the explanations they saw, and the "active user." For example:

> S1MB5: *"It made me think of it like how the AI is thinking. Is it thinking long term? Is it thinking short term? Thinking about the two different lanes each time?... what the best decision would be or what I would make as the decision, so you asking that question made me think 'was my own decision better?'."*
>
> S1MB8: *"...it was good to kind of evaluate myself where I was at when thinking about what decisions the AI was doing, so I can better evaluate the next stage."*
>
> S2MF41: *" Being able to compare the AI's choices in the explanation graphs made it helpful in seeing what may have been a stronger choice (AI vs yourself)."*
>
> S2MB35: *"Friendly AI bought 1 baneling building in bottom lane. I'm unable to notice all possible changes at a decision point."*

One form of metacognition is self-explanation, and our approach encouraged some participants to generate their own explanations:

> S1MB10: *"I think the aim of the AI is to increase the number of minerals, and then go to the last one that is immortals, so that they can make a great damage to the nexus."*
>
> S2MB8: *"Plans to distract Enemy AI in bottom lane."*
>
> S2MF42: *" AI doesn't appear to consider killing bottom lane to be an avenue to victory."*

Finally, while our process promoted thinking about the future, the artifacts also supported participants' ability to reflect on the *past*:

> S1MB19: *"These cards? It's good to write good points and bad points for every three rounds, so that we can go back and see what mistakes we did from the bad."*

4.10 Threats to Validity

Any study has threats to validity, which can skew results towards particular conclusions [200].

One such threat was the participants' amount of domain expertise. Evaluators of an AI system need domain knowledge to evaluate the AI's performance in the domain, and some of the participants may not have had enough RTS experience. As an example, 46% of Study One's participants had at least 10 hours of RTS gaming experience. It is possible that these participants' experience levels may have impacted their ability to evaluate an AI in that domain. Also, it was not clear how to interpret large decreases in the number of clarifications a participant requested early vs. late in the process. It could have meant that the participants understood the explanations over time, or alternatively that they simply gave up. The question wording could also have influenced participants' responses. Many were written and uniformly worded in a balanced set of positive, negative, and neutral wording, but the verbal post-task interview wording was informal, so more subject to individual variation.

The reliability of qualitative coding rests upon inter-rater reliability (IRR) measures. We used Jaccard [77], and 80% is considered good agreement, but for one code set we achieved only 76%. Another hindrance to the generalizability of our findings is the circumscribed design and small size of our study—preventing comparative statistics from yielding meaningful contrast between Study Two's treatments. Similarly, the sensemaking task we have chosen focuses on the *depth* and *breadth* of participants' constructed mental models—but says little to nothing about their *accuracy* or *usefulness*.

Also, qualitative studies are intended to reveal phenomena on approaches that have not been investigated before, and are not suitable for generalization. That said, our study can still inform model-based explanations for domains where the branching factor is small (or can be made small via pruning, as we have done).

4.11 Conclusion

In this paper, we have presented AAR/AI (After-Action Review for AI), a new assessment method to bring accountability to both AI agents and to the humans who must assess them. To inform the design of AAR/AI, we present results from two qualitative lab studies to learn what people need when assessing an AI agent, as well as pros/cons of both the AAR/AI process and the explanations embedded in the process. Among the phenomena we found were:

- *"Organized," "Logical," and..."Repetitive":* Some participants remarked that AAR/AI process helped them think logically and stay organized. Some appreciated its support for reflection on past thoughts. Notably, the process helped participants generate rationale for events with long time lags. However, some bemoaned the repetitiveness of the AAR/AI questions.
- *Explanation complexity:* Our search tree explanations for a model-based agent were approximately the right complexity for some of the participants to understand. They reported being able to *"draw their own conclusions"* from them, and appeared to be using them to align the agent's prediction with the actual future. Other participants did not fully understand the diagram. This mix of attitudes toward the same explanation corroborates other research reporting that explanations are not "one size fits all" (e.g. [11]), and suggests allowing people to access different actions and/or explanation types on demand.
- *Model-Free or Model-Based:* In Study Two we had both Model-Free and Model-Based explanations. Study Two participants who used the Model-Free explanations expressed less than half as many explanation-informed statements as the Model-Based participants did. More critically, the Model-Free participants' bug reports were merely participants' disagreements with the agent's strategy, whereas some Model-Based participants were able to point explicitly to logic errors in the explanations.
- *Diversity of perspectives:* As we observed and participants reported, AAR/AI's questions encouraged participants to consider their observations from multiple, different perspectives, which research suggests may produce problem-solving benefits [52].
- *How many and which:* To answer some of the AAR/AI questions, participants needed to compare items in the explanation from a very large set of options, the sheer quantity of which made them hard to co-locate. We provided the AI's most promising options,

but some participants wanted to see options the AI considered *bad*, as well as actions that spend all resources. Accommodating different people's comparison needs to answer the AAR/AI questions is an unresolved issue—so methods to support scalable comparisons of items in large datasets (e.g. [128]) is an active area of Info Viz research.

- *From whence:* Some participants needed to know the *provenance* of axiomatic values (value estimations at the leaf nodes). That said, if people are to be held accountable for relying on an AI agent, then the ability to "audit" its decision making by allowing the ability to trace provenance may be a requirement.

Overall, AAR/AI's ability to organize participants' work with our agent's explanations assisted the participants in the assessment process. Our results are particularly promising when combining AAR/AI with Model-Based explanations. Still, developing useful explanations and rigorously measuring their quality remains quite difficult and, as our participants pointed out, there is much work still to be done. Ultimately, we hope that AAR/AI's framework around explanations can help people like S1MB14 see *"the flow of logic that we* ***should've*** *had"* when assessing AI systems that impact us daily.

Chapter 5: Addendum to "After-Action Review for AI (AAR/AI)": Finding AI's Faults with AAR/AI: An Empirical Study

By: 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.

Summary of a paper[1] to appear in: ACM Trans. Interact. Intell. Syst.

Abstract: Would you allow this AI agent to make decisions on your behalf? If the answer is "not always", the next question becomes "in what circumstances"? Answering this question requires human users to be able to assess an AI agent—and not just with overall pass/fail assessments or statistics. Here users need to be able to *localize* an agent's bugs, so that they can determine when they are willing to rely on the agent and when they are not. After-Action Review for AI (AAR/AI), a new AI assessment process for integration with Explainable AI systems, aims to support human users in this endeavor, and in this paper, we empirically investigate AAR/AI's effectiveness with domain-knowledgeable users. Our results show that AAR/AI participants not only located significantly *more* bugs than non-AAR/AI participants did (i.e., showed greater recall), they also located them more *precisely* (i.e., with greater precision). In fact, AAR/AI participants outperformed non-AAR/AI participants on every bug and were, on average, almost 6 times as likely as non-AAR/AI participants to find any particular bug. Finally, evidence suggests that incorporating labeling into the AAR/AI process may encourage domain-knowledgeable users to abstract above individual instances of bugs; we hypothesize that doing so may have contributed further to AAR/AI participants' effectiveness.

[1]We later performed a quantitative evaluation, which we briefly summarize here—but omit the full version because it is in the MS thesis for the first author of that paper.

5.1 Methodology

To investigate the effectiveness of the AAR/AI process for localizing AI's faults/bugs, we conducted an empirical study with domain-knowledgeable participants using AAR/AI vs. without AAR/AI. Due to COVID-19, we conducted sessions over teleconference (Zoom) and a browser-based custom combination of the platform (game and explanation system, including AAR/AI features for the AAR/AI treatment) and questionnaires. We required participants to be at least 18 years of age, and to have 10+ hours of prior experience with real-time strategy (RTS) games to ensure they would understand our domain. In addition, we excluded respondents who had taken any AI or ML class before. Of the final 65 participants, 49 self-identified as men, 15 as women, and 1 as transgender. Participants were randomly assigned by flipping a coin to one of two treatments: AAR/AI and non-AAR/AI. Each zoom session had one to seven participants. Upon completing the study, they received a $20 Amazon gift card as compensation.

We used the same agent and domain as in Chapter 4, as well as a modified version of the explanations. Participants' task was to localize the AI agent's bugs. Participants in both the treatments saw the *same explanation*—the only difference between the treatments was the presence/absence of the AAR/AI supports.

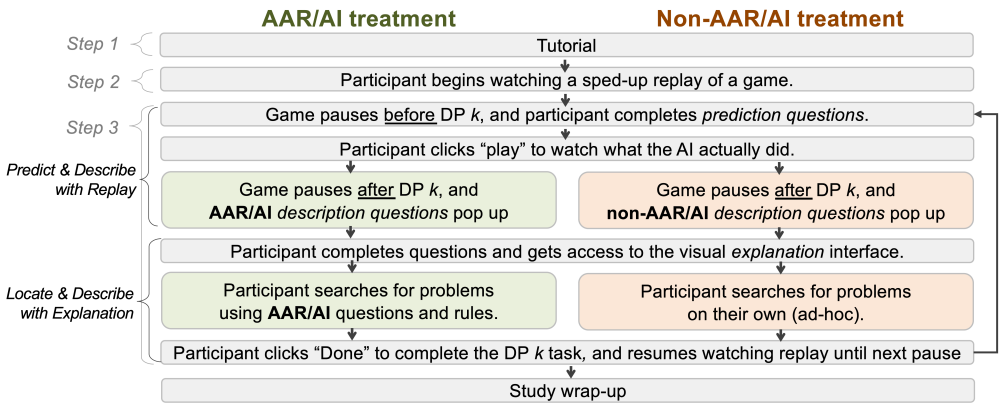


Figure 5.1: Summary of study procedure.

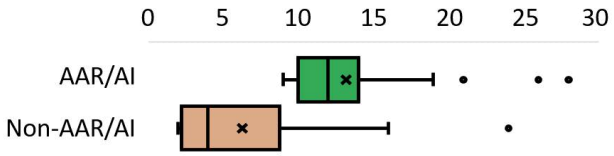


Figure 5.2: Problem report count per participant. AAR/AI: Mean=13.182, SD=4.565; Non-AAR/AI: Mean=6.281, SD=5.050. The AAR/AI participants submitted significantly more problem reports than their Non-AAR/AI counterparts.

Figure 5.3: Participants' recall (left) and precision (right). Recall AAR/AI: Mean=0.233, SD=0.160; Non-AAR/AI: Mean=0.080, SD=0.105. Precision AAR/AI: Mean=0.179, SD=0.129; Non-AAR/AI: Mean=0.116, SD=0.121 over all 10 bugs. AAR/AI participants performed significantly better than Non-AAR/AI participants with both measures.

5.2 Results Summary

To evaluate the correctness of their problem reports, we use the term "bug" to refer to true bugs, and the term "problem" to denote whatever participants reported as problematic. We use these concepts to compute two metrics commonly used in machine learning – recall and precision. Recall measures the proportion of the system's 10 bugs the participants reported, and precision measures the proportion of participants' problem reports that were actually bugs. An "ideal" participant's problem reports would include all bugs (perfect recall) and nothing else (perfect precision).

Using these measures, the AAR/AI participants had both significantly greater recall (Welch's t-test, $t(55.666) = 4.5479$, $p < .0001$) and precision (t-test, $t(63) = 2.0358$, $p = .04598$) than the Non-AAR/AI participants (Figure 5.3). Cohen's d showed a large effect size ($d = 1.121$) for the recall difference, and a medium effect size ($d = .505$) for the precision difference[2].

[2]We consider Cohen's $d \in [0, 0.2)$ to be no effect, $d \in [0.2, 0.5)$ to be small, $d \in [0.5, 0.8)$ to be medium,

Together, these results suggest that the AAR/AI process not only encouraged participants to report significantly more problems (Figure 5.2), it also encouraged them to report problems that were indeed bugs, as measured by their significantly higher recall and precision (Figure 5.3). These results are especially encouraging given that none of the participants had backgrounds in AI/ML.

and $d \in [0.8, 1.4)$ to be large, by convention [32].

Chapter 6: How Do People Rank Multiple Mutant Agents?

By: Jonathan Dodge, Andrew Anderson, Matthew Olson, Rupika Dikkala, and Margaret Burnett

To appear in: 27th International Conference on Intelligent User Interfaces (IUI '22). ACM, New York, NY, USA.
DOI: https://doi.org/10.1145/3490099.3511115

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.

6.1 Introduction

Explaining episodic decisions is a significant challenge with much ongoing XAI research (e.g., [22, 46]). 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. [68]. 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 [39, 92]. 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."* [69].

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 6.1), 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

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

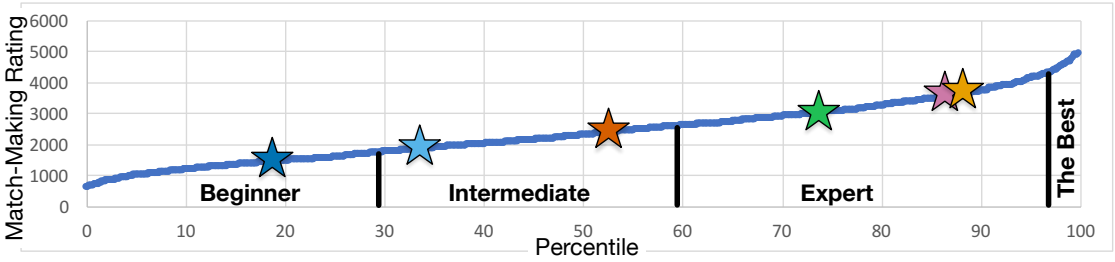


Figure 6.1: Hypothetical Matchmaking Rating (MMR) chart in a game showing the distribution of players' skill, akin to figures from Vinyals et al. [191] or Robertson et al. [150]. The background line is the whole player population, and the stars correspond to the true skill levels of a collection of agents 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.

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 [20, 38, 137].

Using the Ranking Task and Mutant Agent Generation, we conducted a qualita-

tive 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) [45, 88, 114] 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;
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);
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.

6.2 Background

6.2.1 Explanations and Users' Mental Models

Hoffman et al. hypothesized that a *"good mental model will enable [users] to develop appropriate trust in the AI..."* [69]. Those authors enumerated a number of mental model elicitation strategies (detailed in [69]'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 provides outputs [97]. The role of explanations is

to make such boxes less opaque. Explanations have been shown to improve mental models [97, 100], satisfaction (in the colloquial sense, the user's self-reported feeling) [7, 83], and understanding—particularly in low expertise observers [203]. 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 [183].

In order to inform a mental model given potentially complex explanations, including charts and figures, participants may support acquiring transferable knowledge [148] via "self-explanations." In particular, learners employing less successful strategies rely heavily on examples, struggling to self-explain [29]. 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 [50, 93].

6.2.2 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 such as Chess, Shougi, and Go, recently reaching performance exceeding the best humans [166, 164]. As Reeves et al. [147] 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 [192, 191]. Explaining RTS actions remains a challenge, though Metoyer et al. [117] studied how expert-novice pairs do it. More recently, Penney et al. [136] examined how professional commentators and lab participants behaved in an effort to inform explanation design. For example, Madumal et al. [113] 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 [64]. 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") [75]. Another approach from Huang et al. [74] seeks to select states via *criticality* (max_action - average_action). Amir and Amir [10] 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. [207] 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 [187]. Olson et al. [129] analyzed policy trajectories by generating counterfactuals for critical states. Other promising strategies explaining policies globally: extracting automata [198], rules [122], or decision trees [210].

6.2.3 "Testing" AI

AI and ML systems have some important terminological differences from traditional software systems that bear clarifying. For example, Groce et al. [59] 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."* [59]. 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 [56] 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."* [56]. However, this list of interventions assumes the assessor has deep ML/AI knowledge.

In light of the uniqueness of the machine learning pipeline [68], some analysis tools allow the user to inspect each element of the pipeline. One of these, GAN Lab by Kahng et al. [81] is intended for instructional purposes. Another, ModelTracker by Amershi et al. [8], 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 [80]. Other techniques that operate on opaque boxes include: LIME [149], inspection of predictions [95, 112], test

selection [59, 162], and counterfactuals [193].

Interactive tools are a very recent alternative to opaque box explanations. Though they are often published within information visualization literature, they offer some of the more powerful mechanisms for inspecting complicated ML systems. Some are for instructional purposes, (e.g. [81]), but others are state-of-the-art systems [27, 79, 130]. More recently, these techniques have been applied to models for data scientists [70].

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 [49]. 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."* [26]. To address these challenges, researchers have devised a number of strategies, surveyed by Gilpin et al. [55]. One is rendering the network more interpretable while attempting to maintain performance [28, 105]. Another is to *verify* networks, surveyed by Melis et al. [116]. One of the most recent describes DeepTest [180], which attempts to use image processing for data augmentations. These extra images achieve greater "neuron coverage" [134], which is analogous to "code coverage" from software engineering.

6.2.4 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 [101, 97].

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 [159]. The AAR was devised by the U.S. Army [185, 121], but sees continued use [156, 62, 19]. AAR has been adapted for use in other domains, such as medical treatment [157], emergency preparedness [36], fire fighting [76] and AI via a variant known as AAR/AI [45, 114].

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 [114, 45]: 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. [88] 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 [86]. Our experiment includes an adaptation of AAR/AI.

6.3 The Explanations; and the Agents that Generate Them

Section 6.2 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 6.2.

Figure 6.3 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 [33]. The three explanations appear in Figures 6.3, 6.4, and 6.5; each with the same board state and mouse position (yellow highlighted square in Figure 6.3).

6.3.1 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 6.2). 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

[2]Our program is written in Python, with dependencies for GUI elements (wxPython [176]), graphics (OpenGL [163]), and neural networks (PyTorch [133]). Full implementation: `https://github.com/dodgej/RankingMutants`

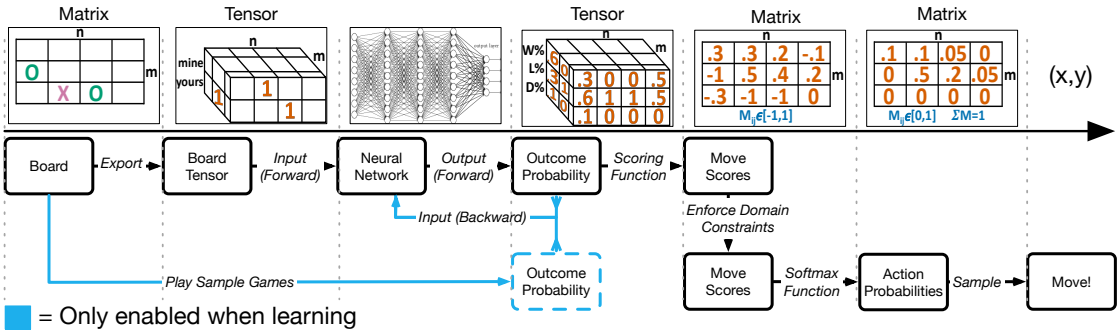


Figure 6.2: How the agent makes a decision, showing nouns in black boxes, verbs on arrows, and the data involved above each step in the pipeline. The process begins with a board, which gets converted to a board tensor, then passed into the CNN. The CNN outputs an outcome probability tensor, which is 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.

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. [173], though still used, e.g., [110]). 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 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

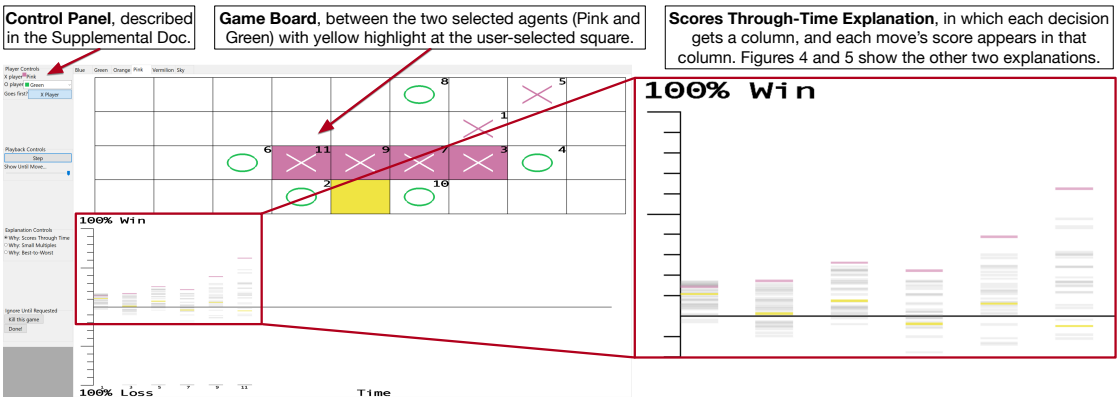


Figure 6.3: The environment, showing the Scores Through-Time explanation (Section 6.3.2). The control panel (left) allowed participants to pick agents, an explanation type to view, and step through the games. The game board (top) shows where X (pink) and O (green) moved, and the highlighted X's show 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 6.4 and 6.5 show the other two explanations.

at the current state, with 10 rollouts[3] 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.

6.3.2 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 Fig-

[3]Note that 10 rollouts is very few, resulting in noisy probability estimates. Despite this, the agent performs fairly well even with this limited amount of computation.

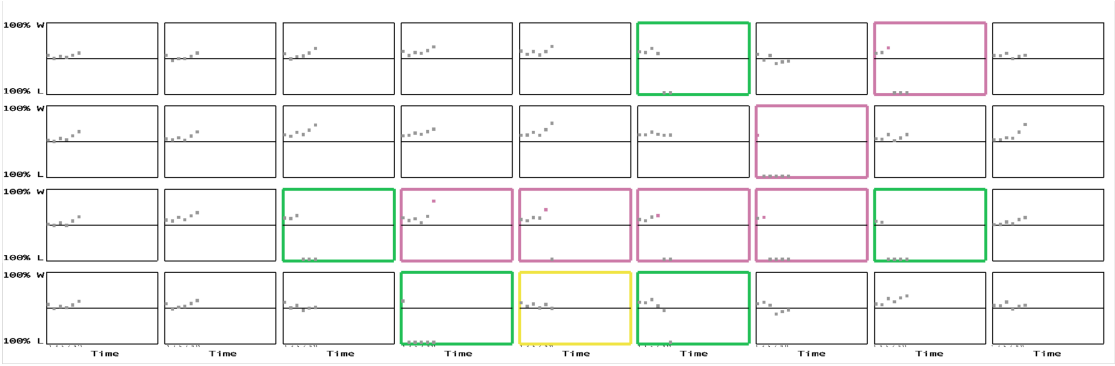


Figure 6.4: The Scores On-the-Board Explanation. Each move gets a small chart of Scores Through-Time, with occupied squares colored by the agent's color (pink and green; yellow indicates the square highlighted in Figure 6.3). Figure 6.3 uses our old name for this explanation.

ure 6.3, the Pink X player's highest-scoring[4] 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 score the agent gave to 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.

6.3.3 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

[4]Since we left the Gumbel-softmax on, the policy was probabilistic, meaning if a score with very slightly lower score might occasionally become the agent's choice. This design decision kept games from being deterministic once the user chose a pair of agents.

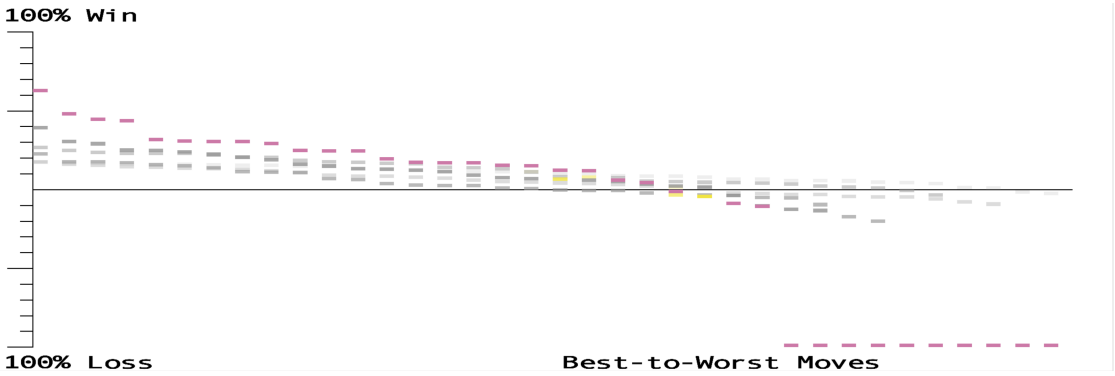


Figure 6.5: The Scores Best-to-Worst Explanation. Each decision results in a single sorted data series, which are identified by color (pink is the most recent, then grey colors from dark to light).

grid element is an *StTime* chart for one decision, intended to ease comparison [182]. For example, the top left chart in Figure 6.4 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.

6.3.4 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.

6.4 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?

After receiving IRB approval, we recruited participants by posting flyers around the community. All our participants gender identified[5] 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

[5]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 [158]

associated with a branch of the military.

6.4.1 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 [164].

Other researchers have used MNK (e.g., [4]), partially because *"people's intuitive priors (three-in-a-row is good) happen to be correct"* [188]. While Go has a similar representation, MNK rules are much simpler, making it well-suited for HCI studies.

6.4.2 Manipulating agent "quality": Mutant Agent Generation

In Section 6.1, 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 6.1a). 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 $mean = 0$

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]
#5Agent					[909, 79, 12]

(b) Upper diagonal of matchup matrix, showing results from Table 6.1a broken down per pair of agents.

Table 6.1: Ground truth, results from large round-robin tournament.

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 6.1a; each step down the ranking equals ≈700 fewer wins).

6.4.3 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 6.1b), as in Kim et al. [90]. 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 6.3 shows accessible colors from [201]).

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.

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 [114, 45]. 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[6] 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 6.1a 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).

6.5 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[7], 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 6.1a).

Table 6.2 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

[6]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).

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

	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 6.2: 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.

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 this the "pigeonhole score," per the mathematical pigeonhole principle [67]. Each participant's pigeonhole score is the number of empty cells per column in Table 6.2.

While Table 6.2 emphasizes pigeonhole success *by participant*, Figure 6.6 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 6.1—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."*

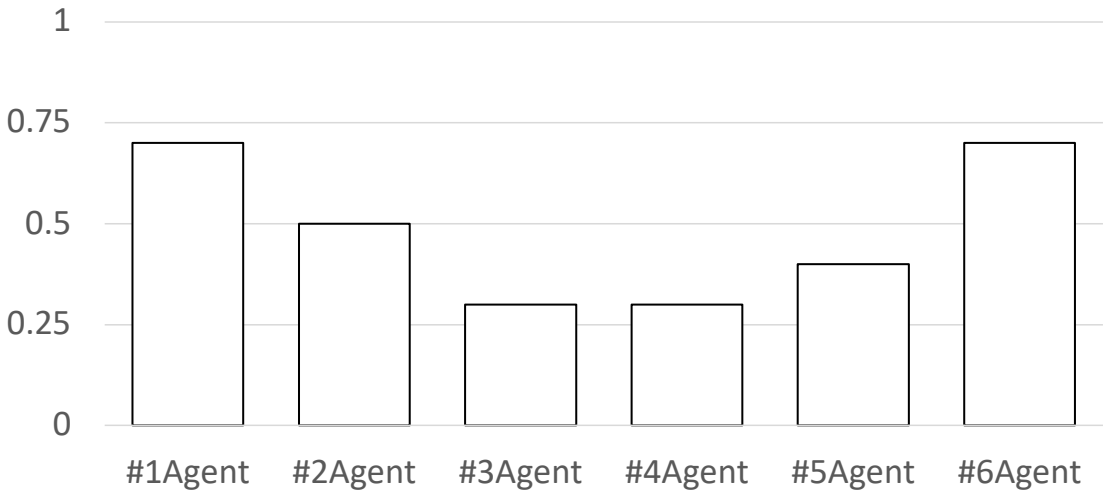


Figure 6.6: The fraction of participants (y-axis) who correctly ranked each agent. The U shape points out how much more successful participants were with the top/bottom agents than with the middle agents.

6.6 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 6.7 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 6.7 shows, P04 and P05 gave only token glances at explanations other than *StTime*.

6.6.1 Participants' Explanation Diets

We can view participants' explanation choices through Information Foraging Theory's [139] 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

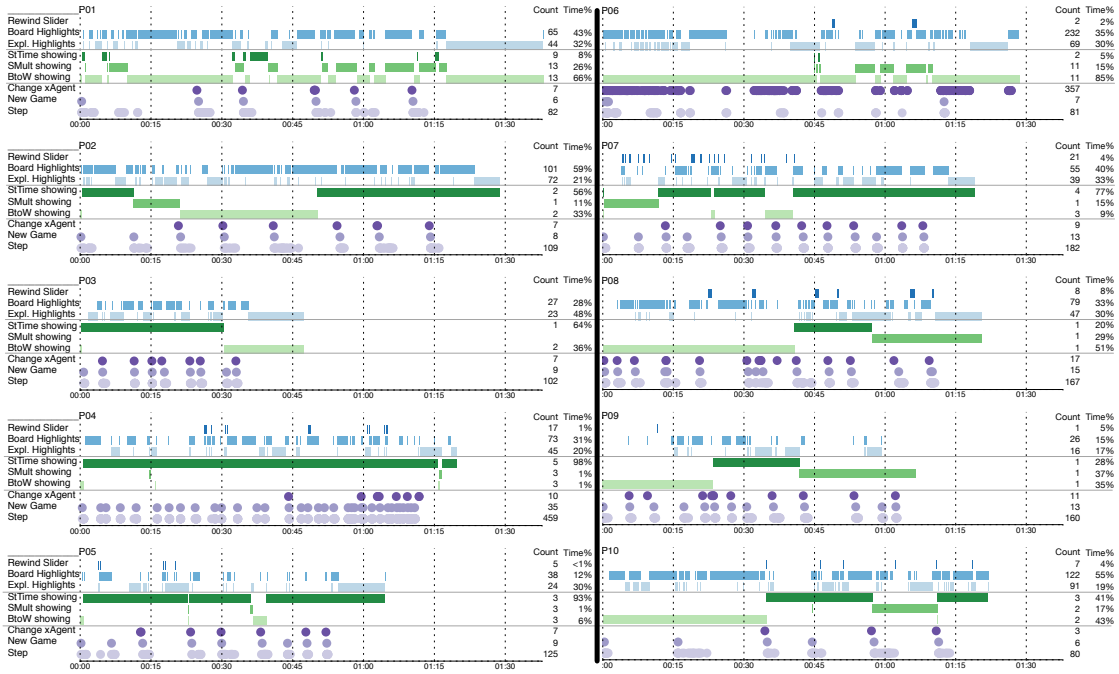


Figure 6.7: Timelines of each participant's events, with minutes into the main task on the X-axis. The top 3 rows (blue) show participants' interactions with the explanation. The middle 3 rows (green) show which explanation is currently visible. The bottom 3 rows (purple) show participants' interactions 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%).

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 [138]. Figure 6.8 summarizes participants' usage patterns that were detailed in Figure 6.7, and both starkly reveal the Repeat diet pattern for P04 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:

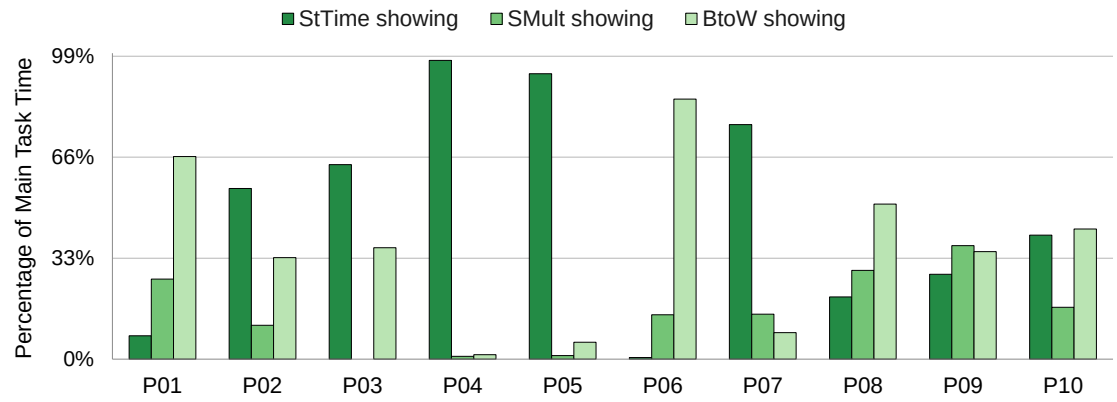
> P04: *"[#1Agent]'s gauge of win probability is flawed. It could guarantee a win earlier."*

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

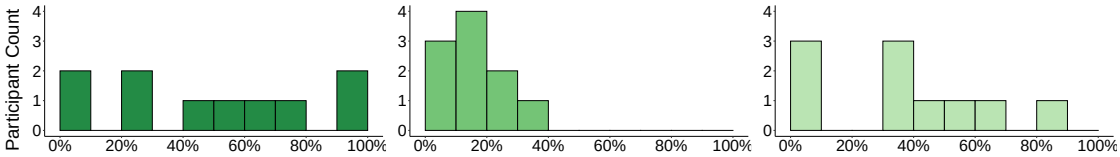
We term a second diet pattern, apparent in both Figure 6.7 and Figure 6.8, the *Serial Repeat* dietary pattern. In this pattern, a participant would remain with the same explanation type for a fairly long period of time (at least 10 minutes) before switching to another explanation type, where they would 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 6.7. Figure 6.8a shows that each of these five participants spent $\geq 25\%$ of their time in one explanation type and $\geq 25\%$ 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 6.7). 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."*



(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 6.8: Charts of participant usage behaviors for each explanation type.

6.6.2 Which explanation types?

Figure 6.8b 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.

6.6.2.1 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.

> 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 6.9 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 6.3) attempts to show the win probability for *all* actions, as opposed to just the one that occurred.

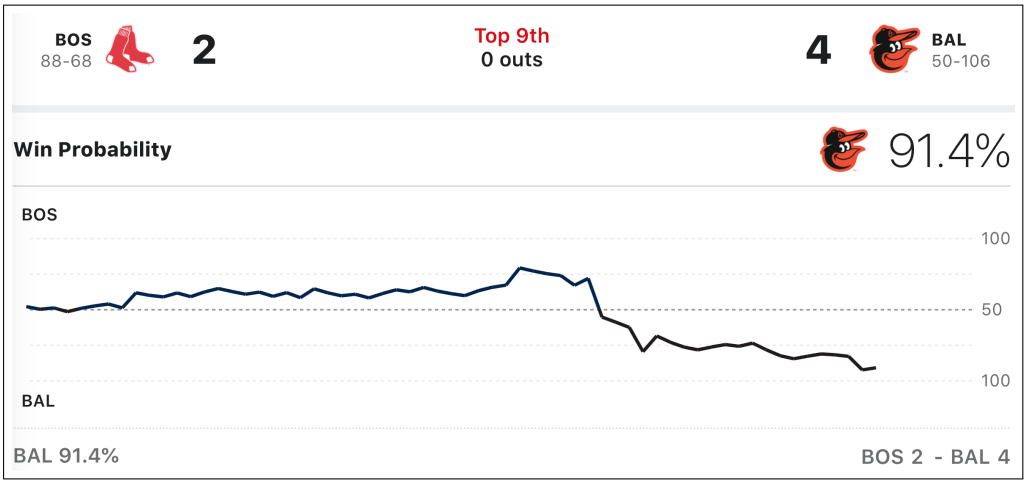


Figure 6.9: This ESPN chart shows analysts' estimates of two baseball teams' win probability over time, sampled near the end of the game (Source: `https://www.espn.com/mlb/game/_/gameId/401229397`). P04 said the *StTime* explanation felt familiar, due to similarity with sports charts like this.

6.6.2.2 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."*

6.6.2.3 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 6.7).

> 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."*

6.6.3 Implications for Interactive XAI and 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. [12], 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 [96]) may be a useful strategy to reduce this empirical cost.

6.7 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 6.10 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 $\otimes$ beneath the chart). However, some participants, 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 $\bigcirc$ 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 $\times$ beneath the chart).

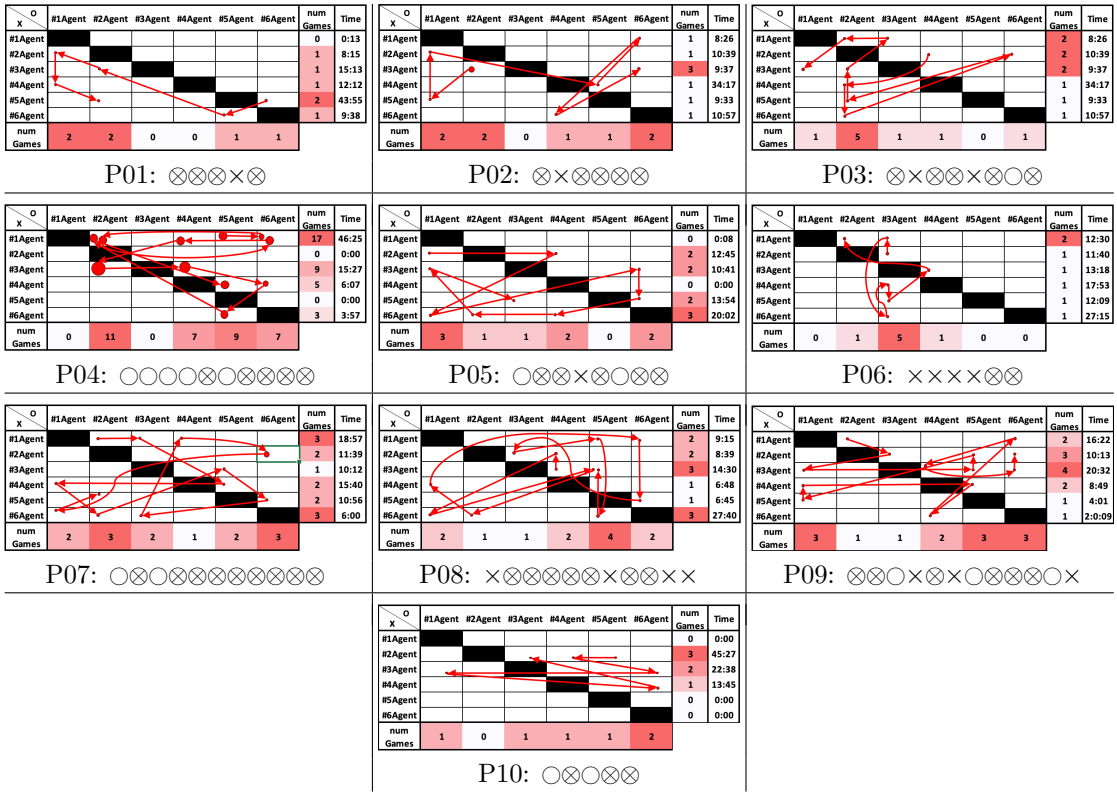


Figure 6.10: How participants selected agents to assess throughout the main task, expressed as the red path through the matchup matrices, as expressed by the "New Game" data on Figure 6.7. Participants' path steps in these matrices are: × (vertical in the matrix) changes the X-agent. ○ (horizontal) changes the O-agent. ⊗ (diagonal) changes both agents. Size of the red dot reflects how long (# games) a participant stayed with the same pairing.

6.7.1 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 6.7)—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.

6.7.2 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 6.10 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."*

6.7.3 "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 6.11). 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.

> 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.

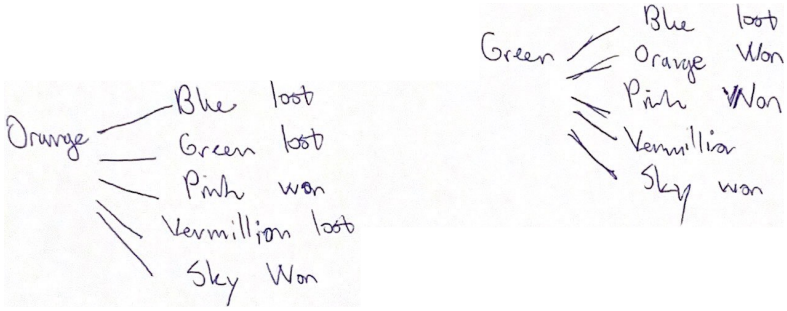


Figure 6.11: Two of six stumps drawn by P09 to assist in finalizing ranking. Each of these was initially created after P09 had generated a hypothesized final ranking, and written in that order. Then, using this artifact, P09 selected a few more agent pairs to assess before declaring the task complete.

6.7.4 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 6.4), 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 6.5 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

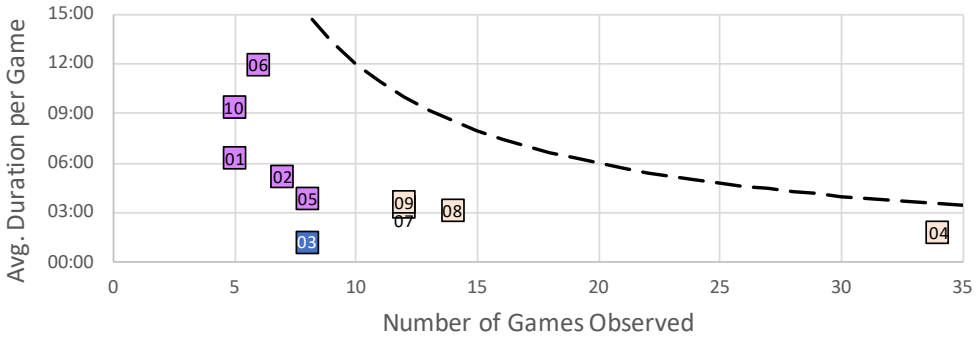


Figure 6.12: The number of games each participant observed (x-axis) and how long they watched each game on average (y-axis). The numbers are their PID numbers. Since they had 2 hours to complete their task, participants had to generate strategies to maximize the value of the information they received 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.

a way to track their progress. One possibility would be to include a matchup matrix in the interface similar to Figure 6.10, 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.

6.8 Results RQ4: How did participants invest their time while ranking?

Filling the matchup matrix from Section 6.7 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.

6.8.1 Invest in Many Games

ManyGames participants spent less time per game ($\leq Median(Avg.\ Duration)$) so as to see more games ($> Median(\#\ Games)$) (Figure 6.12) . 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[8] 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 #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."*

6.8.2 Invest Thoroughly in Games

Five participants spent more time per game[9] ($> Median(Avg.\ Duration)$) but in turn saw fewer games ($\leq Median(\#\ Games)$). *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*

[8] In Table 6.1b, an underdog victory is where a lower-ranked agent wins against a higher-ranked opponent (i.e., #3Agent defeating #1Agent).

[9] 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).

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%"*

6.8.3 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 [143] 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?

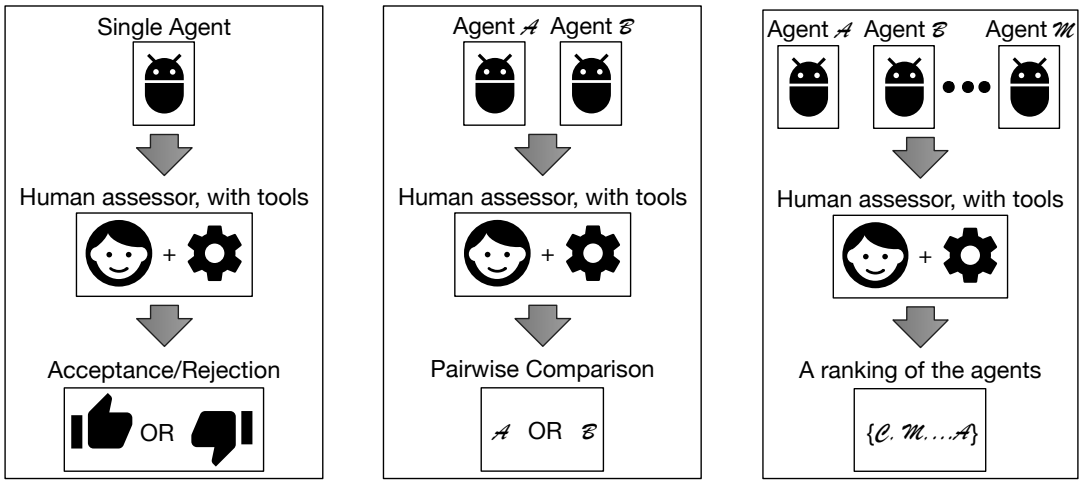
To illustrate, while we previously highlighted how forthcoming *ThoroughGames* investors seemed as research participants, we had some difficulty obtaining high quality written data from *ManyGames* investors. Concretely, only 1 of 4 *ManyGames* investors had a written response we coded as Explanation 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.

6.9 Discussion

6.9.1 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. [69].

For example, Anderson et al. [12] 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 [44]. Prediction tasks can benefit from "partial credit" for when



(a) Acceptance testing: Provided *one* input item, assessors determine fitness *"for the purpose."* [61].

(b) Comparison testing: Provided *two* input items (as in [80]), assessors determine "which is better."

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

Figure 6.13: 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).

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 6.13 illustrates measuring advantages the Ranking Task brings to XAI researchers. Acceptance testing (left) [61] is challenging to ground truth, as it can be difficult to define criteria for the assessor's acceptance. Comparison testing (middle) [80] 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[10] 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 6.14 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?

6.9.1.1 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

[10]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.

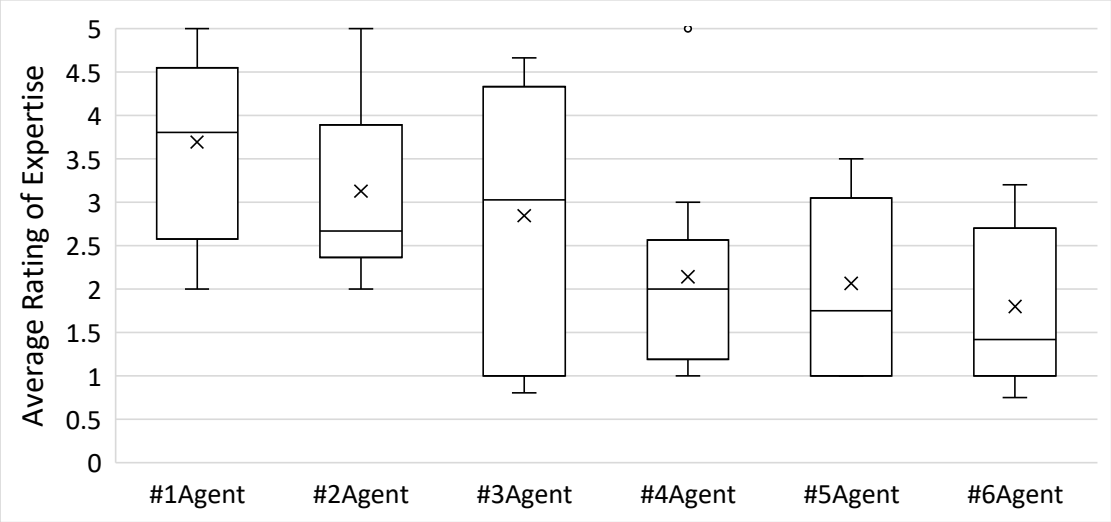


Figure 6.14: Distributions of participants' average ratings (y-axis) of the 6 agents' expertise (x-axis). Notice that in Figure 6.6, participants' accuracy ranking the agents was U-shaped, but participants' in-situ ratings of the agents expertise here was more aligned with the agent quality (i.e., participants rated 6-MNL worst, and #1Agent best).

binning participants into an explanation-type treatment if they used that type at least half as often as that participant's most-used. Table 6.3 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.)

6.9.2 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.

"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 6.3: 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.

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 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[11] 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 [120]) bear significant resemblance to reports found in various sports[12], so perhaps these two communities can learn from each other.

6.9.3 The Ranking Task vs. AutoML

One might argue that automating the application of ML to real world problems, (AutoML [178]) reaches straight for the large scale past-facing evaluation data that we use

[11] *"In sociology and science and technology studies, a boundary object is information, such as specimens, field notes, and maps, used in different ways by different communities for collaborative work through scales."* Source: `https://en.wikipedia.org/wiki/Boundary_object`

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

for "ground truth" (e.g., Figure 2 in Wang et al. [194] 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., [15] 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 [127]. 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).

6.9.4 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 [38], but still used today [137]. 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 $\geq$ 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[13]; while large amounts of noise will produce an essentially random agent. Table 6.1a illustrates that the most damaged agent is on par with a random one, and that "Low" noise agents are

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

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. [75, 74] used this strategy, finding it assisted human assessment by selecting more informative states. Other properties researchers have manipulated include opaqueness [141], complexity [141], fairness [46], 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 [144]. 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[14] with little development effort[15], similar to how mutation testing can be applied in semi-automated ways [160] to essentially any source code.

6.10 Threats to Validity

Every study has threats to validity [200]—in reviewing ours, we follow Yin's approach [205].

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 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. [131]). 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 [136]. Lastly, although good agents exist [191], they are not publicly available. The best alternative agents are heterogeneous competition

[14]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.

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

submissions (as in Kim et al. [90]), making explanation difficult. Meanwhile, the compute required to train quality StarCraft agents is infeasible for most researchers [40].

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 [74].

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.

6.11 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 6.5). Fortunately, researchers can measure this quantity to reveal where an explanation type is (in)adequate (Section 6.9.1.1).
- ...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."

Chapter 7: Addendum to "How Do People Rank Multiple Mutant Agents?": Measuring Explanation Resolution

7.1 Introduction

After the qualitative study described in Chapter 6, we adapted our interface and collected more data. Among our changes, we enabled the interface to show all three explanations at once. During Chapter 6's study, participants could freely choose the explanation to view and for how long—but only one at a time. We had already observed that some participants frequently switched between explanations, suggesting they were using explanations in *combination* with each other.

To explore explanation combinations, we recruited 87 participants to perform our Ranking Task in-lab. During the task, the software presented participants with 0–3 of our explanations (Scores Through-Time, Scores On-the-Board, Scores Best-to-Worst), resulting in the following combinations (treatments): *StTime*; *OnBoard*; *BtoW*; *StTime+OnBoard*; *StTime+BtoW*; *OnBoard+BtoW*; *ALL three explanations*; *NO explanations*. We also experimentally controlled participants' time usage, via time boxing with a short password dialog to give each participant a consistent amount of time per game.

7.2 Computing Explanation Resolution

Now, with these empirical controls, we re-run the computation demonstrated in Section 6.9.1.1's case study, using the large scale game results found in Appendix O as ground truth. Figure 7.1, Figure 7.2, and Table 7.1 present our results, which show a few surprising trends.

First, we see that our two top performing treatments (as measured by participants' agent rankings) are both explanation *pairs* (*StTime+OnBoard*, *OnBoard+BtoW*), suggesting that the combination of two explanations helped participants answer different

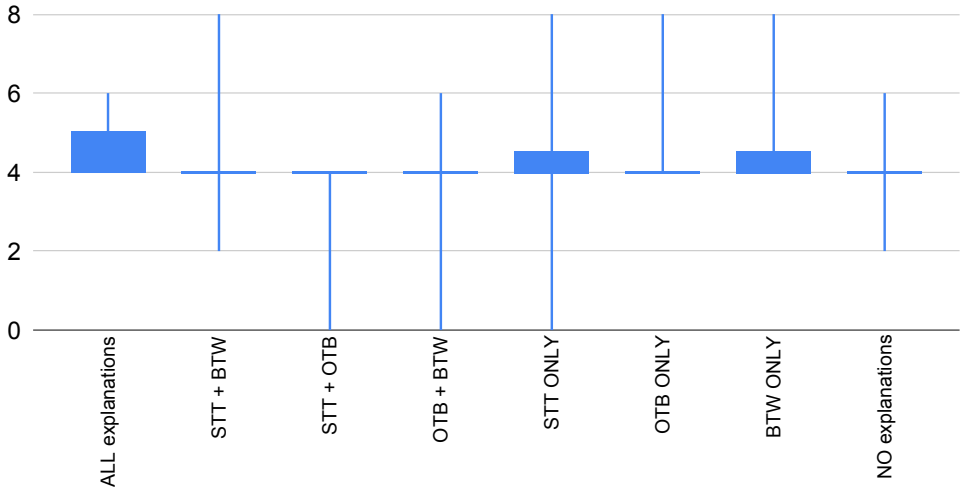


Figure 7.1: Ranking Loss per Treatment, so lower is better. Participants ranked 4 agents, leading to possible losses of {0,2,4,8}. As Appendix O demonstrates, this ranking task was fairly difficult, as the agents did not have a large difference between them.

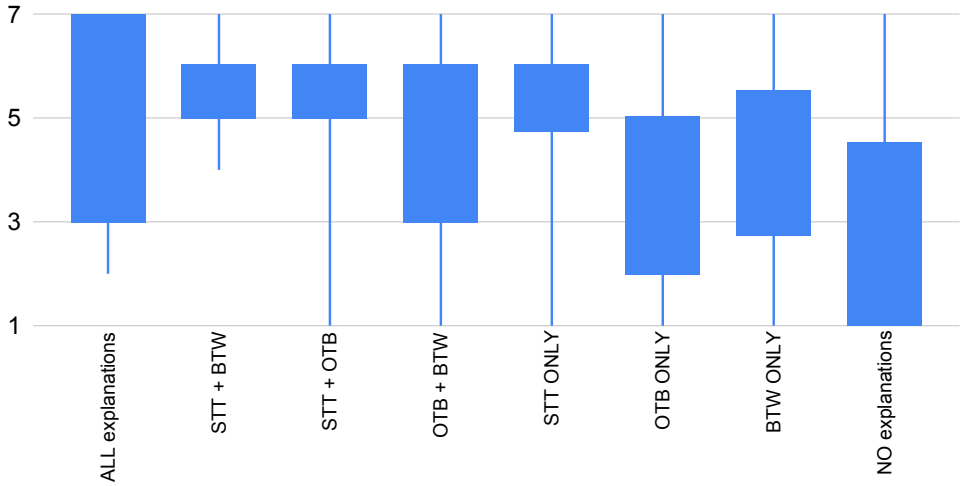


Figure 7.2: Participants' Perception Score after doing the Ranking Task, indicating *"How do you feel about each EXPLANATION COMBINATION?"* (Likert scores, 'Like very much'=7 to 'Dislike very much'=1)

	Ranking Loss		Perception Score	
Treatment	Mean	Ordered	Mean	Ordered
StTime+BtoW	4.40	5	5.56	1
StTime+OnBoard	3.60	1	5.20	2
StTime	4.36	4	5.08	3
ALL three explanations	4.55	6	4.82	4
OnBoard+BtoW	3.80	2	4.44	5
BtoW	4.60	7	4.00	6
OnBoard	4.73	8	3.91	7
NO explanations	3.82	3	2.91	8

Table 7.1: Calculation of explanation resolution, which we quantify via Ranking Loss, for an ablation of our three explanations. Next to resolution, we show how popular the explanation was with our participants (detailed in Figure 7.2). Since the metrics are opposite measures, we added "ordered" columns where 1 is best.

questions. *NO explanations* closely followed our two top performing treatments. After these three treatments, there is a sizable gap in mean ranking loss, and the rest of the treatments have similar performance scores. It's interesting in that the solo explanations that are components of one the better performing pairs (*OnBoard+BtoW*) were both terrible alone, see Figure 7.1 and Table 7.1). Our results again emphasize that trying to devise and empirically show the best (solo) XAI explanation is probably not the right goal.

From the perception scores, it is clear that people did *not* like using *NO explanations*, as it comes in a full point below the other options. Despite their dislike of the *NO explanations* treatment, participants performed surprisingly well with only the board state. Also, we see that again, participants seem to perceive *pairs* of explanations better than individual ones (pairs are in positions 1, 2, 5, while solo is in 3, 5, 7, and *NO explanations* is below *ALL three explanations*). Here and in Chapter 6, nobody liked *OnBoard* much alone, but it turned out to be a very helpful complement to either or *StTime* or *BtoW*. This is especially evident in performance (Figure 7.1 but also somewhat echoed in perception (Figure 7.2.

Chapter 8: Future Directions and Concluding Remarks

The primary components of my future XAI research plans will be to investigate four research topics that appear central to XAI: (1) how Explanation Resolution, in conjunction with tasks like Ranking [42], can improve researchers' ability to evaluate explanations; (2) how mutant generation [42] on AI/ML systems can help measure the quality of testing frameworks, including explanations; (3) how Explanation Templates can contribute to systematically measuring explanation quality [44]; and (4) how human evaluation *processes* combine with the systems that embed explanation. Along the way, we should seek to make (X)AI available to many people, and much more cheaply. Currently, (X)AI suffers from high capital investment costs and the classism that follows [40]. Hopefully, some of the approaches described in this dissertation can help illuminate a brighter future.

The first topic is to explore the extent to which the Resolution framework described in Step 6 (Chapter 6) improves researchers' ability to measure the effectiveness of explanations. In particular, resolution could be applied to many existing explanation artifacts and processes, but we have only explored resolution in a single study. Thus, there is a lot of opportunity available in measuring old techniques with our new ruler. However, the goal of measurement is not for its own sake—it should inform improved designs.

The second topic is to apply different kinds of mutant generation to different kinds of AI/ML techniques. The advantage of mutant testing in software engineering comes from the fact that it adds essentially no constraints on the underlying source code. In Chapter 6 we discussed how additive noise can be applied to neural networks in a similar way. While semantically meaningful perturbations might be superior for testing purposes, they will become domain/task/agent specific. However, other perturbation approaches (e.g., multiplicative noise) could be used in a similarly constraint-free way, and on other kinds of approaches.

The third topic is to explore the extent to which **Templates** can assist in measuring explanation quality [44]. Templates are like the factory that generates the explanation. However, that is not their only power; just as factories can receive inspections and

improvements, so can templates. In this way, templates can improve how explanation designers systematically consider the large space of possible design decisions without having to rely on human inspection of individual explanation *instances*—which may be too numerous to tractably examine.

The fourth topic is to continue exploring human assessment **Processes**, e.g., AAR/AI (Step 5). We have just begun to explore this space, and I would like to investigate how to improve and generalize it in various ways, e.g., across domains, at different action granularities, etc. Of particular import is to offer *enough* structure each assessor is not forced to re-invent their own assessment process; but not *so much* structure that each assessor is constrained to enact similar processes. Some users may have a process in mind, and it will be important to offer them a system that allows them the freedom to enact *their* process. To provide such flexibility, one strategy might be to offer highly configurable interfaces (e.g., multiple panels where the user can choose from a variety of possible explanations in each panel and set the number of panels).

The main thrust of this research was creating useful, novel XAI technologies and enabling humans to understand and quantify the effects of such technologies. Human user evaluation continues to be an understudied area of XAI—Keane and Kenny [84] surveyed **1,102** papers on "post-hoc explanation by example" and observed: *"In all the papers we examined we found less than a handful (i.e., <**5**) that performed any adequate user testing of the proposal that cases improved the interpretability of models; this gap needs to be rectified."* This indicates that *many* researchers are proposing explanations, *few* are evaluating their proposals with users, and *even fewer* are thinking about the process those users might follow.

In the future, the success of AI-based systems is likely to depend on multidisciplinary study of a variety of topics, including Artificial Intelligence, Human-Computer Interaction, Software Engineering, Visualization, Social Sciences, and more. Without combining these myriad fields, it will be impossible to evaluate the impact of such systems by observing their effects over time on users and society as a whole.

Bibliography

[1] Artificial intelligence's white guy problem, 2016. Accessed: 6/8/2018.

[2] Rise of the racist robots–how AI is learning all our worst impulses, 2017. Accessed: 6/8/2018.

[3] COMPAS recidivism risk score data and analysis, 2018. Accessed: 6/8/2018.

[4] Abdel-Hafiz Abdoulaye, Vinasetan Ratheil Houndji, Eugène C. Ezin, and Gael Aglin. Generic heuristic for the MNK-games. In *African Conference on Research in Computer Science*, CARI '18, pages 265–275, 10 2018.

[5] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 582. ACM, 2018.

[6] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9525–9536. Curran Associates, Inc., 2018.

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

[8] Saleema Amershi, Max Chickering, Steven Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. ModelTracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2015)*. ACM - Association for Computing Machinery, April 2015.

[9] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Dan Amir and Ofra Amir. HIGHLIGHTS: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and*

MultiAgent Systems, pages 1168–1176. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[11] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett. Explaining reinforcement learning to mere mortals: An empirical study. In *International Joint Conference on Artificial Intelligence*, Macau, China, 10–18 August 2019. IJCAI.

[12] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett. Explaining reinforcement learning to mere mortals: An empirical study. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 1328–1334, Palo Alto, CA, USA, 2019. AAAI Press.

[13] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Matthew Olson, Alan Fern, and Margaret Burnett. Mental models of mere mortals with explanations of reinforcement learning. *ACM Trans. Interact. Intell. Syst.*, 10(2), May 2020.

[14] Lorin W. Anderson, David R. Krathwohl, Peter W. Airasian, Kathleen A. Cruikshank, Richard E. Mayer, Paul R. Pintrich, James Raths, and Merlin C. Wittrock. *A Taxonomy for Learning, Teaching, and Assessing: A revision of Bloom's Taxonomy of Educational Objectives*. Pearson, New York, NY, USA, 2001.

[15] Nathan Benaich and Ian Hogarth. State of AI report, 2020.

[16] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 377:1–377:14, New York, NY, USA, 2018. ACM.

[17] Benjamin S. Bloom, Max D. Engelhart, Edward J. Furst, Walker H. Hill, and David R. Krathwohl. *Taxonomy of Educational Objectives*. Longmans, Green and Co LTD, London, England, 1956.

[18] Tim Brennan, William Dieterich, and Beate Ehret. Evaluating the predictive validity of the COMPAS risk and needs assessment system. *Criminal Justice and Behavior*, 36(1):21–40, 2009.

[19] Ralph Brewer, Anthony Walker, E. Ray Pursel, Eduardo Cerame, Anthony Baker, and Kristin Schaefer. Assessment of manned-unmanned team performance: Comprehensive After-Action Review technology development. In *2019 International*

Conference on Human Factors in Robots and Unmanned Systems, AHFE '19, pages 119–130, Cham, CHE, 2019. Springer Nature Switzerland AG.

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

[21] John T Cacioppo, Richard E Petty, and Chuan Feng Kao. The efficient assessment of need for cognition. *Journal of personality assessment*, 48(3):306–307, 1984.

[22] Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 258–262, New York, NY, USA, 2019. ACM.

[23] Carrie J. Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. "hello AI": Uncovering the onboarding needs of medical practitioners for human-AI collaborative decision-making. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), November 2019.

[24] Toon Calders and Indrė Žliobaitė. *Why Unbiased Computational Processes Can Lead to Discriminative Decision Procedures*, pages 43–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[25] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3992–4001. Curran Associates, Inc., 2017.

[26] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2016.

[27] Nan-Chen Chen, Jina Suh, Johan Verwey, Gonzalo Ramos, Steven Drucker, and Patrice Simard. AnchorViz: Facilitating classifier error discovery through interactive semantic data exploration. In *Proceedings of the 23th International Conference on Intelligent User Interfaces*, pages 269–280. ACM, March 2018.

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

[29] Michelene T.H. Chi, Miriam Bassok, Matthew W. Lewis, Peter Reimann, and Robert Glaser. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182, 4 1989.

[30] William J Clancey. The epistemology of a rule-based expert system–a framework for explanation. *Artificial intelligence*, 20(3):215–251, 1983.

[31] CNN. Who's responsible when an autonomous car crashes?, 2016.

[32] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[33] Michael Correll, Dominik Moritz, and Jeffrey Heer. *Value-Suppressing Uncertainty Palettes*, page 1–11. Association for Computing Machinery, New York, NY, USA, 2018.

[34] Kelley Cotter, Janghee Cho, and Emilee Rader. Explaining the news feed algorithm: An analysis of the "news feed FYI" blog. In *ACM CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1553–1560. ACM, 2017.

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

[36] Robert Davies, Elly Vaughan, Graham Fraser, Robert Cook, Massimo Ciotti, and Jonathan E. Suk. 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):618–625, june 2019.

[37] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13:319–340, 1989.

[38] R. A. DeMillo, R. J. Lipton, and F. G. Sayward. Hints on test data selection: Help for the practicing programmer. *Computer*, 11(4):34–41, April 1978.

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

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

[41] Jonathan Dodge. Position: The Case Against Case-Based Explanation. In *IUI Workshops*, 2022.

[42] Jonathan Dodge, Andrew Anderson, Matthew Olson, Rupika Dikkala, and Margaret Burnett. How do people rank multiple mutant agents? In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, IUI '22, New York, NY, USA, 2022. ACM. To Appear.

[43] Jonathan Dodge and M. Burnett. Position: We can measure XAI explanations better with templates. In *ExSS-ATEC@IUI*, 2020.

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

[45] 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. After-action review for AI (AAR/AI). *ACM Transactions on Interactive Intelligent Systems*, 2021. (To Appear).

[46] Jonathan Dodge, Q. Vera Liao, Yunfeng Zhang, Rachel K. E. Bellamy, and Casey Dugan. Explaining models: An empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 275–285, New York, NY, USA, 2019. ACM.

[47] Jonathan Dodge, Sean Penney, Claudia Hilderbrand, Andrew Anderson, and Margaret Burnett. How the experts do it: Assessing and explaining agent behaviors in real-time strategy games. In *2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 562:1–562:12, New York, NY, USA, 2018. ACM.

[48] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O. Riedl. Automated rationale generation: A technique for explainable ai and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 263–274, New York, NY, USA, 2019. ACM.

[49] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Xiaodong Song. Robust physical-world attacks on deep learning visual classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

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

[51] Donna-Lynn Forrest-Pressley and GE MacKinnon. *Metacognition, Cognition, and Human Performance: Theoretical Perspectives*, volume 1. Academic Pr, 1985.

[52] Hershey H Friedman, Linda W Friedman, and Chaya Leverton. Increase diversity to boost creativity and enhance problem solving. *Psychosociological Issues in Human Resource Management*, 4(2):7, 2016.

[53] GDPR. European union general data protection regulation, article 15 - "right of access by the data subject", 2018. Accessed: 1/16/2019.

[54] Peter Gerjets, Katharina Scheiter, and Richard Catrambone. Designing instructional examples to reduce intrinsic cognitive load: Molar versus modular presentation of solution procedures. *Instructional Science*, 32(1-2):33–58, 2004.

[55] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 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*, pages 80–89, 2018.

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

[57] Ian Goodfellow and Nicolas Papernot. The challenge of verification and testing of machine learning, 2017.

[58] Nina Grgic-Hlaca, Elissa M. Redmiles, Krishna P. Gummadi, and Adrian Weller. Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 903–912, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[59] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Transactions on Software Engineering*, 40(3):307–323, March 2014.

[60] Sara Hajian, Francesco Bonchi, and Carlos Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2125–2126. ACM, 2016.

[61] Brian Hambling and Pauline van Goethem. *User acceptance testing: a step-by-step guide*. BCS Learning and Development, Swindon, 2013.

[62] Samer Hanoun and Saeid Nahavandi. Current and future methodologies of after action review in simulation-based training. In *2018 Annual IEEE International*

Systems Conference (SysCon), SysCon '18, pages 1–6, New York, NY, USA, 2018. IEEE.

[63] S. G. Hart and L. E. Staveland. Development of NASA-TLX (task load index): results of empirical and theoretical research. *Adv. Psychol.*, 52:139–183, 1988.

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

[65] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[66] Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. Assessing acceptance of assistive social agent technology by older adults: the Almere model. *International Journal of Social Robotics*, 2(4):361–375, Dec 2010.

[67] IN Herstein. Topics in algebra-walthan, 1969.

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

[69] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for explainable AI: challenges and prospects. *CoRR*, abs/1812.04608, 2018.

[70] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. 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*, CHI '19, pages 579:1–579:13, New York, NY, USA, 2019. ACM.

[71] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–16, New York, NY, USA, 2019. Association for Computing Machinery.

[72] Hsiu-Fang Hsieh and Sarah E Shannon. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9):1277–1288, 2005.

[73] Ming Hsieh and Shi-Chun Tsai. On the fairness and complexity of generalized -in-a-row games. *Theor. Comput. Sci.*, 385:88–100, 10 2007.

[74] Sandy H. Huang, Kush Bhatia, Pieter Abbeel, and Anca D. Dragan. Establishing appropriate trust via critical states. *IROS*, Oct 2018.

[75] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. Enabling robots to communicate their objectives. *CoRR*, abs/1702.03465, 2017.

[76] Andrew Ishak and Elizabeth Williams. Slides in the tray: How fire crews enable members to borrow experiences. *Small Group Research*, 48(3):336–364, March 2017.

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

[78] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

[79] Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng Chau. ActiVis: Visual exploration of industry-scale deep neural network models. *IEEE Transactions on Visualization and Computer Graphics*, 24:88–97, 2018.

[80] Minsuk Kahng, Dezhi Fang, and Duen Horng (Polo) Chau. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA '16, pages 1:1–1:6, New York, NY, USA, 2016. ACM.

[81] Minsuk Kahng, Nikhil Thorat, Duen Horng (Polo) Chau, Fernanda B. Viégas, and Martin Wattenberg. GAN lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Trans. Vis. Comput. Graph.*, 25(1):310–320, 2019.

[82] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

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

[84] Mark T. Keane and Eoin M. Kenny. How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In Kerstin Bach and Cindy Marling, editors, *Case-Based Reasoning Research and Development*, pages 155–171, Cham, 2019. Springer International Publishing.

[85] Nathanael Keiser and Winfred Arthur, Jr. A meta-analysis of the effectiveness of the After-Action Review (or debrief) and factors that influence its effectiveness. *Journal of Applied Psychology*, 08 2020.

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

[87] Caitlin Kelleher and Wint Hnin. Predicting cognitive load in future code puzzles. In *2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 257:1–257:12, New York, NY, USA, 2019. ACM.

[88] 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. Finding AI's faults with AAR/AI: An empirical study. *ACM Transactions on Interactive Intelligent Systems*, 2021. To Appear.

[89] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288, 2016.

[90] M. Kim, K. Kim, S. Kim, and A. K. Dey. Performance evaluation gaps in a real-time strategy game between human and artificial intelligence players. *IEEE Access*, 6:13575–13586, 2018.

[91] Man-Je Kim, Kyung-Joong Kim, SeungJun Kim, and Anind K Dey. Evaluation of starcraft artificial intelligence competition bots by experienced human players. In *ACM CHI Conference Extended Abstracts*, pages 1915–1921. ACM, 2016.

[92] Alexandra Kirsch. Explain to whom? putting the user in the center of explainable AI. In *CEx@AI*IA*, 2017.

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

[94] Rafal Kocielnik, Saleema Amershi, and Paul N. Bennett. Will you accept an imperfect AI? exploring designs for adjusting end-user expectations of AI systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[95] Josua Krause, Adam Perer, and Kenney Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5686–5697, New York, NY, USA, 2016. ACM.

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

[97] T. Kulesza, M. Burnett, W. Wong, and S. Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *ACM International Conference on Intelligent User Interfaces*, pages 126–137. ACM, 2015.

[98] T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W. K. Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10, Sept 2013.

[99] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *ACM Conference on Human Factors in Computing Systems*, pages 1–10. ACM, 2012.

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

[101] Todd Kulesza, Weng-Keen Wong, Simone Stumpf, Stephen Perona, Rachel White, Margaret M Burnett, Ian Oberst, and Andrew J Ko. 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*, pages 187–196, 2009.

[102] Adam Lareau and Brice Long. The art of the After-Action Review. *Fire Engineering*, 171(5):61–64, May 2018.

[103] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm, 2016. Accessed: 6/8/2018.

[104] Min Kyung Lee. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society*, 5(1):2053951718756684, 2018.

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

[106] Q Vera Liao and Wai-Tat Fu. Expert voices in echo chambers: effects of source expertise indicators on exposure to diverse opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2745–2754. ACM, 2014.

[107] Brian Lim, Anind Dey, and Daniel Avrahami. *Why* and *Why Not* explanations improve the intelligibility of context-aware intelligent systems. In *2009 SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2119–2128, New York, NY, USA, 2009. ACM.

[108] Brian Y Lim. *Improving understanding and trust with intelligibility in context-aware applications*. PhD thesis, Carnegie Mellon University, 2012.

[109] Brian Y Lim, Anind K Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2119–2128. ACM, 2009.

[110] Zhengxian Lin, Kin-Ho Lam, and Alan Fern. Contrastive explanations for reinforcement learning via embedded self predictions. In *International Conference on Learning Representations*, 2021.

[111] Sandra Deacon Lloyd Baird, Phil Holland. Learning from action: Imbedding more learning into the performance fast enough to make a difference. 27:19–32, 1999.

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

[113] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. *CoRR*, abs/1905.10958, 2019.

[114] 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. Keeping it "organized and logical":

After-Action review for AI (AAR/AI). In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 465–476, New York, NY, USA, 2020. Association for Computing Machinery.

[115] 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. Keeping it "organized and logical": After-Action Review for AI (AAR/AI). In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 465–476, New York, NY, USA, 2020. Association for Computing Machinery.

[116] Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *ICLR*. OpenReview.net, 2018.

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

[118] Tim Miller. Explanation in artificial intelligence: insights from the social sciences. *arXiv preprint arXiv:1706.07269*, 2017.

[119] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, New York, NY, USA, 2019. Association for Computing Machinery.

[120] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, New York, NY, USA, 2019. Association for Computing Machinery.

[121] John E. Morrison and Larry L. Meliza. Foundations of the After Action Review Process. Technical report, Institute for Defense Analyses, 1999.

[122] W. James Murdoch and Arthur Szlam. Automatic rule extraction from long short term memory networks. *ArXiv*, abs/1702.02540, 2017.

[123] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *CoRR*, abs/1802.00682, 2018.

[124] Donald A Norman. Some observations on mental models. *Mental Models*, 7(112):7–14, 1983.

[125] N.Y. Times. Tesla's self-driving system cleared in deadly crash, 2017.

[126] N.Y. Times. Wielding rocks and knives, Arizonans attack self-driving cars, 2018. Accessed: 1/16/2019.

[127] Johan Samir Obando-Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1373–1383. PMLR, 2021.

[128] Oluwakemi Ola and Kamran Sedig. Beyond simple charts: Design of visualizations for big health data. *Online journal of public health informatics*, 8, 12 2016.

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

[130] Matthew L. Olson, Thuy-Vy Nguyen, Gaurav Dixit, Neale Ratzlaff, Weng-Keen Wong, and Minsuk Kahng. Contrastive identification of covariate shift in image data. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 2021.

[131] S. Ontañón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss. A survey of real-time strategy game ai research and competition in StarCraft. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(4):293–311, Dec 2013.

[132] Giuliano Orru and Luca Longo. The evolution of cognitive load theory and the measurement of its intrinsic, extraneous and germane loads: a review. In *International Symposium on Human Mental Workload: Models and Applications*, pages 23–48. Springer, 2018.

[133] 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. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[134] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. DeepXplore. *Proceedings of the 26th Symposium on Operating Systems Principles - SOSP '17*, 2017.

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

[136] Sean Penney, Jonathan Dodge, Claudia Hilderbrand, Andrew Anderson, Logan Simpson, and Margaret Burnett. Toward foraging for understanding of StarCraft agents: An empirical study. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 225–237, New York, NY, USA, 2018. ACM.

[137] Goran Petrović and Marko Ivanković. State of mutation testing at google. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '18, pages 163–171, New York, NY, USA, 2018. ACM.

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

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

[140] Karl R Popper. Science as falsification. *Conjectures and refutations*, 1:33–39, 1963.

[141] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. *Manipulating and Measuring Model Interpretability*. Association for Computing Machinery, New York, NY, USA, 2021.

[142] John Quarles, Samsun Lampotang, Ira Fischler, Paul Fishwick, and Benjamin Lok. Experiences in mixed reality-based collocated After-Action Review. *Virtual Reality*, 17(3):239–252, September 2013.

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

[144] Ivet Rafegas, Maria Vanrell, LuÃs A. Alexandre, and Guillem Arias. Understanding trained CNNs by indexing neuron selectivity. *Pattern Recognition Letters*, 136:318–325, 2020.

[145] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7(Aug):1655–1686, 2006.

[146] Stephen Reed, Alexandra Dempster, and Michael Ettinger. Usefulness of analogous solutions for solving algebra word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(1):106–125, January 1985.

[147] Stuart Reeves, Barry Brown, and Eric Laurier. Experts at play: Understanding skilled expertise. *Games and Culture*, 4(3):205–227, 2009.

[148] Alexander Renkl, Robin Stark, Hans Gruber, and Heinz Mandl. Learning from worked-out examples: The effects of example variability and elicited self-explanations. *Contemporary Educational Psychology*, 23(1):90–108, January 1998.

[149] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*, pages 1135–1144. ACM, 2016.

[150] Justus Robertson, Athanasios Vasileios Kokkinakis, Jonathan Hook, Ben Kirman, Florian Block, Marian F Ursu, Sagarika Patra, Simon Demediuk, Anders Drachen, and Oluseyi Olarewaju. Wait, but why?: Assessing behavior explanation strategies for real-time strategy games. In *26th International Conference on Intelligent User Interfaces*, IUI '21, page 32–42, New York, NY, USA, 2021. Association for Computing Machinery.

[151] Ariel Rosenfeld, Moshe Cohen, Matthew E. Taylor, and Sarit Kraus. Leveraging human knowledge in tabular reinforcement learning: a study of human subjects. *The Knowledge Engineering Review*, 33:e14, 2018.

[152] Ariel Rosenfeld, Matthew E. Taylor, and Sarit Kraus. Leveraging human knowledge in tabular reinforcement learning: A study of human subjects. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3823–3830, 2017.

[153] Quentin Roy, Futian Zhang, and Daniel Vogel. Automation accuracy is good, but high controllability may be better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–8, New York, NY, USA, 2019. Association for Computing Machinery.

[154] Stuart Russell and Andrew Zimdars. Q-decomposition for reinforcement learning agents. In *Intl. Conf. on Machine Learning*, pages 656–663, 2003.

[155] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[156] Margaret Salter and Gerald Klein. After Action Reviews: Current observations and recommendations. Technical report, U.S. Army Research Institute for the Behavioral and Social Sciences, 2007.

[157] Taylor Lee Sawyer and Shad Deering. Adaptation of the US Army's After-Action Review for simulation debriefing in healthcare. *Simulation in Healthcare*, 8(6):388–397, December 2013.

[158] Morgan Klaus Scheuerman, Katta Spiel, Oliver L Haimson, Foad Hamidi, and Stacy M Branham. HCI guidelines for gender equity and inclusivity. *UMBC Faculty Collection*, 2020.

[159] Martin Schindler and Martin J Eppler. Harvesting project knowledge: a review of project learning methods and success factors. *International Journal of Project Management*, 21(3):219 – 228, 2003.

[160] David Schuler and Andreas Zeller. 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*, ESEC/FSE '09, page 297–298, New York, NY, USA, 2009. Association for Computing Machinery.

[161] Burr Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1467–1478. Association for Computational Linguistics, 2011.

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

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

[164] 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. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[165] 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. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[166] 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. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[167] Dag IK Sjøberg, Tore Dybå, Bente CD Anda, and Jo E Hannay. Building theories in software engineering. In *Guide to advanced empirical software engineering*, pages 312–336. Springer, 2008.

[168] Alison Smith-Renner, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. Digging into user control: Perceptions of adherence and instability in transparent models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 519–530, New York, NY, USA, 2020. Association for Computing Machinery.

[169] Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning–perspectives and goals. *Artificial Intelligence Review*, 24(2):109–143, 2005.

[170] Dan "Artosis" Stemkoski. AlphaStar - Analysis by Artosis. `https://www.youtube.com/watch?v=_YWmU-E2WFc`, 2019.

[171] Simone Stumpf, Vidya Rajaram, Lida Li, Margaret Burnett, Thomas Dietterich, Erin Sullivan, Russell Drummond, and Jonathan Herlocker. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91. ACM, 2007.

[172] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, 67(8):639–662, 2009.

[173] Richard Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick Pilarski, Adam White, and Doina Precup. Horde : A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction categories and subject descriptors. volume 2, 01 2011.

[174] William R Swartout. Explaining and justifying expert consulting programs. In *Computer-assisted medical decision making*, pages 254–271. Springer, 1985.

[175] John Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4):295–312, 1994.

[176] Hugues Talbot. WxPython, a GUI toolkit. *Linux J.*, 2000(74es):5–es, June 2000.

[177] The StarCraft II Community. Tutorials - Sc2MapsterWiki. `https://sc2mapster.gamepedia.com/Tutorials`, 2019.

[178] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 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*, KDD '13, page 847–855, New York, NY, USA, 2013. Association for Computing Machinery.

[179] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars, 2017.

[180] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, pages 303–314, New York, NY, USA, 2018. ACM.

[181] Alan B Tickle, Robert Andrews, Mostefa Golea, and Joachim Diederich. The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1068, 1998.

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

[183] Joe Tullio, Anind K Dey, Jason Chalecki, and James Fogarty. 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*, pages 31–40. ACM, 2007.

[184] Jos W.H.M. Uiterwijk. Solving strong and weak 4-in-a-row. In *2019 IEEE Conference on Games (CoG)*, pages 1–8, 2019.

[185] U.S. Army. Training circular 25-20: A leader's guide to After-Action Reviews. Technical report, Department of the Army, Washington D.C., USA, 1993.

[186] Kristen Vaccaro, Dylan Huang, Motahhare Eslami, Christian Sandvig, Kevin Hamilton, and Karrie Karahalios. The illusion of control: Placebo effects of control settings. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[187] Jasper van der Waa, Jurriaan van Diggelen, Karel van den Bosch, and Mark A. Neerincx. Contrastive explanations for reinforcement learning in terms of expected consequences. *CoRR*, abs/1807.08706, 2018.

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

[189] Michael Veale, Max Van Kleek, and Reuben Binns. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 440:1–440:14, New York, NY, USA, 2018. ACM.

[190] Oriol Vinyals. DeepMind and Blizzard open StarCraft ii as an ai research environment, 2017.

[191] 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. Grandmaster level in StarCraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

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

[193] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. 2017.

[194] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. Human-AI collaboration in data science: Exploring data scientists' perceptions of automated AI. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), November 2019.

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

[196] Qianwen Wang, Yao Ming, Zhihua Jin, Qiaomu Shen, Dongyu Liu, Micah J. Smith, Kalyan Veeramachaneni, and Huamin Qu. ATMSeer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery.

[197] Franz Emanuel Weinert and Rainer H Kluwe. Metacognition, motivation, and understanding. 1987.

[198] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5247–5256, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[199] Michael R Wick and William B Thompson. Reconstructive expert system explanation. *Artificial Intelligence*, 54(1):33–70, 1992.

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

[201] Bang Wong. Points of view: Color blindness. *Nature Methods*, 8:441, May 2011.

[202] Allison Woodruff, Sarah E. Fox, Steven Rousso-Schindler, and Jeffrey Warshaw. A qualitative exploration of perceptions of algorithmic fairness. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA, 2018. Association for Computing Machinery.

[203] Robert H Wortham, Andreas Theodorou, and Joanna J Bryson. Improving robot transparency: real-time visualisation of robot AI substantially improves understanding in naive observers. In *IEEE RO-MAN 2017*, August 2017.

[204] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[205] Robert K. Yin. *Case Study Research: Design and Methods (Applied Social Research Methods)*. Sage Publications, fourth edition. edition, 2008.

[206] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180. International World Wide Web Conferences Steering Committee, 2017.

[207] Tom Zahavy, Nir Ben Zrihem, and Shie Mannor. Graying the black box: Understanding DQNs. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1899–1908. JMLR.org, 2016.

[208] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[209] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[210] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. DeepRED – rule extraction from deep neural networks. In Toon Calders, Michelangelo Ceci, and Donato Malerba, editors, *Discovery Science*, pages 457–473, Cham, 2016. Springer International Publishing.

APPENDICES

Appendix A: Helpful/Problematic code set for Chapter 4

Code: Description	**Example**	**#**
Explanation Overall Quality: Participant found explanation useless or helpful in a vague sense, or in determining reasons for actions in the decision process (clarity, or lack thereof).	S1MB2: *"I think it's pretty easy to understand, like, after looking at for a little while."*	8
Diagram Color Coding: Participant comments on the manner in which an explanation object is colored.	S1MB17: *"The color coding is okay, it's pretty distinctive. I don't know if the background is gray or...even the marines are gray... it was confusing because if it was different color..."*	4
Changing Diagram Data Contents: Participant talks about changing data in the diagram (such as changing the node definitions, changing the key, etc). This is NOT about showing an action/state node that is not present.	S1MB18: *"How much minerals it has, something like that. I would like that to be represented on the diagram."*	7
Diagram Node Contents: Participant wants the diagram to contain more/fewer nodes, (e.g. interactively expand a node, request a specific action be examined, or have a "wider/narrower" tree) OR thinks it contains the right amount.	S1MB11: *"I would just have more options available, you know. ... So sometimes, there are missing... missing options which should be taken."*	16
Diagram Glyph Presentation: Participant comments on the glyphs for the action or state nodes, referring to the way the state information is presented in the glyph	S1MB10: *"As the number of units goes on increasing, the line goes on increasing. And that is why it's short. That's clear, but vertical lines... if it would have been 1 <line>, it would have been great."*	6

Table A.1: Helpful/Problematic code set for the *explanations*. Frequencies are from Study One's post task three questions centered on the explanation and its contents. (*"What was helpful about the information given to you?"*, *"What was problematic about the information given to you?"*, and *"Under what circumstances is the agent likely to make bad decisions?"*)

Appendix B: Design evolution of explanations in Chapter 4 and 5

In Figure B.1, the root node (region 1) shows the current game state and its estimated value. One layer down (region 2) shows the 4 best actions available to the friendly AI in the current game state–and their values, as estimated by the agent based on the tree expansion. The third level of the tree (region 3) shows actions available to the opponent—again, the 4 best actions and their values as estimated by the agent. The fourth level of the tree (region 4) shows the *predicted* state that the agent thinks will ensue based on the current state, taken together with the simultaneous actions from itself and the opponent. From that level, the agent performs another round of search in the same way, resulting in an agent that looks ahead 2 rounds. Each node is shown with the state or action that node depicts, alongside the estimated value of that state/action, shown in more detail in Figures B.2a and B.3a. If that value is part of the principal variation (colloquially, the most likely trajectory given "optimal" play from both sides), its value is shown in green instead of blue.

Figure B.1 depicts the explanation used in Study One. For Study Two, we used the explanations shown in Figure 4.2. These Study Two explanations were implemented in an interactive prototype, hence offering interactions not possible in Study One's paper-prototyped explanations. Also, drawing from our observations of Study One participants, in Study Two's explanations we changed the glyphs used to represent states and actions, including outcome bars, which are shown in more detail in Figures B.2b and B.3b. Based on Study One's results, we made the default tree more complex by increasing the branching factor at the root from 4 to 5, but eliminated the branching between friendly and enemy actions, instead including only the option estimated to be the best.

We improved the explanation in other ways between the studies. For example showing an estimation of the resources available to both the friendly AI and its opponent, as requested by a Study One participant:

> S1MB20: *"I would enjoy to see ... the AI's, calculation of their minerals. ...further extrapolation of getting this many more minerals allows you to buy these units. ...Because in RTS games you think about the enemy's resources as*

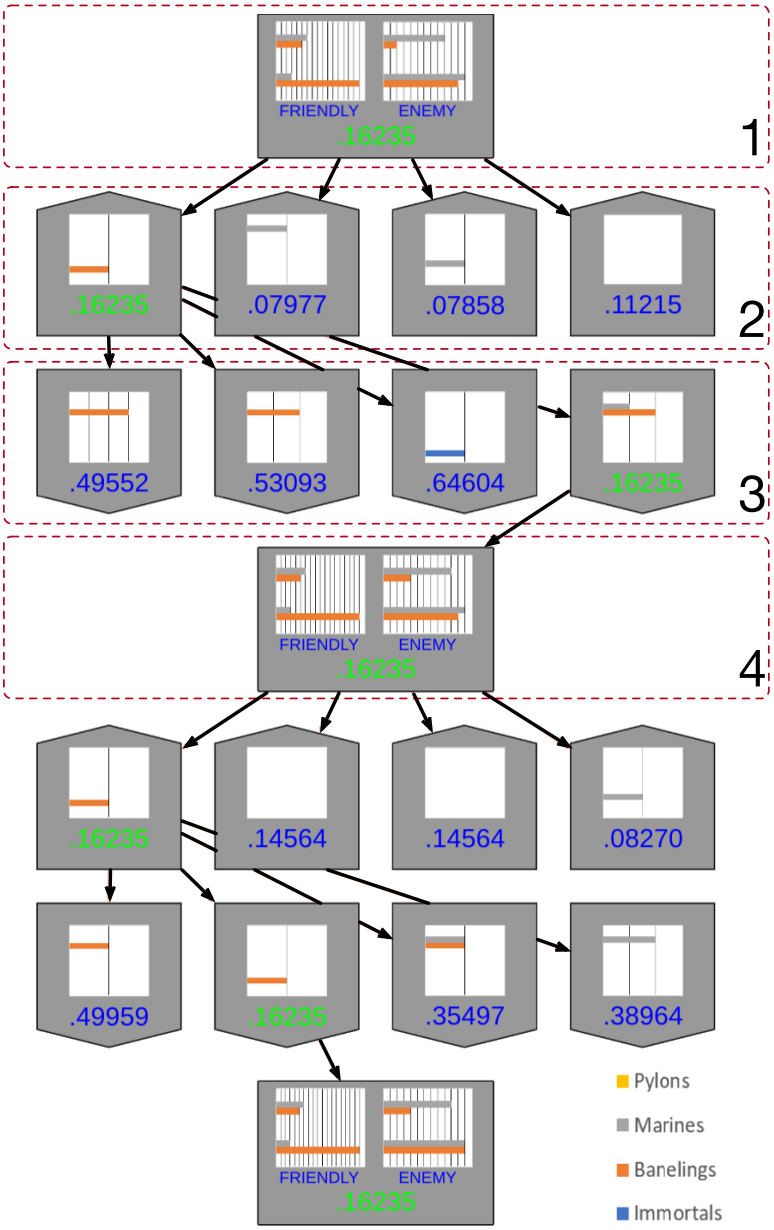


Figure B.1: Search tree explanation for decision point 22 in Study One, presented to participants as a paper prototype. Dashed red boxes show: *(1)* game state at decision point 22, *(2)* top 4 most rewarding actions, as estimated by the AI, *(3)* top 4 most rewarding actions *for the enemy* in response to its "best" action, as estimated by the AI, and *(4)* predicted game state at decision point 23. Our agent searches to depth 2, so the explanation includes another turn of search from the *predicted* state (box 4). Note that all states below the root (box 1) are predicted by the agent. Green highlighted numbers indicate parts of the principal variation.

well and how to manage those as well as your own."

For Study Two, we incorporated much of this Study One feedback into our explanation design, but Study Two participants were not entirely satisfied. Some wanted information that still went beyond that available in the explanations:

S2MB30: *"It would have been helpful to know how many immortals are effective against a baneling and number of marines, effective against immortals etc. Instead of which ones are effective against each other."*

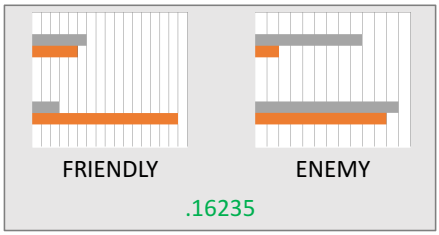
S2MB28: *"There was no info on what enemy AI is thinking. Also both lanes play at same time so hard to focus on both."*

These quotes suggest that finding, processing, and sorting out high-level information intermingled with low-level information was cognitively burdensome. Adding to this cognitive burden, some Study Two participants pointed to the cognitive work of comprehending certain glyphs and layout:

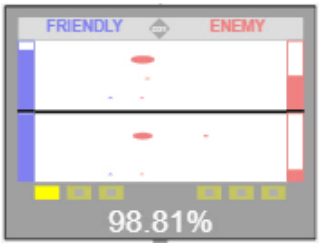
S2MB35: *"I didn't explore all parts of the explanations. Couldn't relate shapes with marines, banelings, or immortal buildings."*

S2MF38: *"Object shapes and names were pretty hard to remember, should have simplified to basic code shapes used in the explanation. Damage powers weren't displayed."*

S2MB26: *"The position of the boxes are wide apart so it takes time to visually go from one box to another. Keeping track of both lanes wide apart is difficult."*

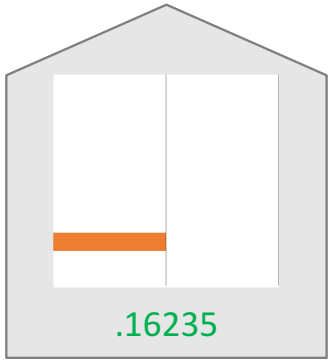


(a) Study One example of State node presentation.

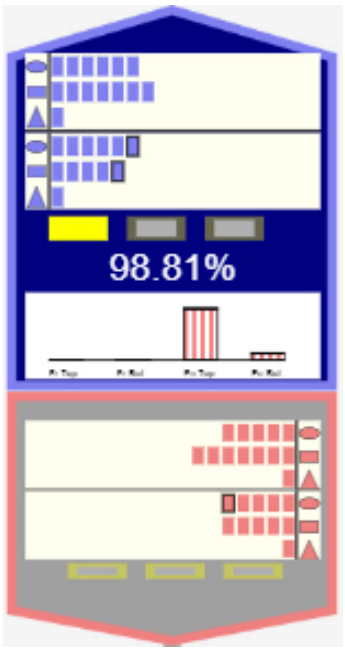


(b) Study Two example of State node presentation.

Figure B.2: In Study One (left), we represented the state with a bar showing a number of unit production facilities for each lane and type. Here, the Friendly AI has 6 marines (gray bar) and 5 banelings (orange bar) in the top lane—with 3 marines and 16 banelings bottom. In Study Two (right), we improved the state representation by including nexus health information via the bars at the edges, as well as pylon count with the yellow/grey rectangles along the bottom. Also the state node, instead of showing troop production facilities, now shows troops that are on the map. This is presented by dividing each lane evenly into four parts, each containing a single shape (oval, square, or triangle) for each type of troop, whose size reflects the number of troops in that part.



(a) Study One example of Action node presentation.



(b) Study Two example of Action node presentation.

Figure B.3: In Study One (left), we used a design similar to states, with bars split by lane and by unit. Each node gives the agent's estimate of the win probability associated with that action (number at the bottom.) In Study Two (right), we improved the action node representation by including both the friendly (top, blue outline) and enemy actions (bottom, red outline) and which lane they are in, with total troop production facilities shown in each lane, and newly acquired production facilities bordered in black. The stacked bar chart illustrates the AI's expectations for likely game outcomes. Each bar shows a nexus's probability of causing a player to lose, with the bar's texture indicating *why* that nexus causes a loss (being destroyed, having lowest health at game end).

Appendix C: More about the CNN Agent from Chapters 6 and 7

C.1 More about the probabilistic policy

Our goal with the probabilistic policy was to prevent games from being deterministic. In particular, during training, the probabilistic property can be used for exploration, but in many applications after training would be replaced with a regular `arg_max` operator. However, doing this in our setting would mean that once the user chose a pair of agents, every game between them would play out the same. To avoid this, we trained our agents using a `Gumbel_softmax`[1] with a temperature parameter, which does not change the *ordering* of the perceived action quality—just the sampling probabilities. We scheduled the temperature to start at 20 and end at 0.1—and continued to use the same 0.1 temperature after training. Our team chose the temperature ranges empirically, running softmax tests directly on hypothetical scoring output instead of using the neural network. The criterion for which we were looking was a sparse action probability matrix, where just a few actions received all the probability mass. To illustrate the importance of sparsity, suppose 35x poor actions have 2% action probability while a single excellent action has all the rest of the probability mass; that would only be 30%!

C.2 Achieving a more optimal agent

Our goal was not to solve the domain, but to get a sufficiently strong agent that we could mutate that agent and see if participants could detect the change. Thus, we employed a simplified version of the architecture outlined in AlphaZero[2], but modified for explanation generation. While our agent is clearly suboptimal, it begs the question of whether its poor performance is the fault of the architecture, the training process, our simplifications, etc. A CNN model should be able to perfectly solve this domain,

[1] `https://pytorch.org/docs/stable/generated/torch.nn.functional.gumbel_softmax.html#torch.nn.functional.gumbel_softmax`

[2] David Silver, et al. 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`

since the game structure is highly local, grid based, and fully described by a single game snapshot. The architecture of our CNN should have sufficient capacity.

Probably the weakest link in the current structure is the target formation. Currently, we use a procedure best described as "Pure Monte Carlo game search" (PMCGS, described by Russel and Norvig[3] in Chapter 5): For each square on the current board, play random games to the end while recording results. Pack the results into a tensor of outcome probabilities (e.g. if it wins 7 of the 10 games, 0.7 would be the win outcome target value). One simple improvement would be to replace the *random games* with games played by having both players *follow the agent's policy*. However, the downside of policy rollouts is that training takes longer. In fact, the capability to do so is currently in our source, but commented out. One of the downstream consequences of using random rollouts is that the agent does not defend very well. This is because if the agent exposes itself to a kill shot, PMCGS targets can underestimate the threat because both players are treated as random, and thus *unlikely* to take the winning square.

The other obvious weak link is the loss function. We attempted to use the simple, off-the-shelf components where possible, and so did not define our own loss function, but it should be possible to improve upon `L1Loss`. In particular, when using L1Loss, it is just as important to accurately estimate an illegal move, a coin flip move, and the winning move; but it is actually far more important to get the last one right. This is not desirable, because e.g., large errors for terrible moves are tolerable without harming the action selection. Conversely, accurate predictions for the best actions seem much more important than the others if the action selection is to pick them. To do so, one could define a custom loss function assigning weight as an increasing function of win probabilities in the *target* tensor.

The last idea that bears mention is to make the agent model-based. Currently, our agent does not get or learn the game model, and so it is essentially model-free. Performing MCTS on the search tree from a model-based agent would likely improve performance, and might be necessary for extending this work into some other domains—but should not be necessary to optimally solve MNK games. However, explaining with this kind of search tree (which explicitly encodes an overwhelming amount of information about the sequential environment) remains an interesting and sparsely explored research area.

[3]Stuart J. Russell and Peter Norvig. 2003. Artificial Intelligence: A Modern Approach (2 ed.). Pearson Education.

To summarize the ideas we suggest trying:

- Use a transformer
- Add an LSTM
- Train longer
- Improve the target formation
- Define a better loss function
- Use a model-based agent

Appendix D: What domain to study? StarCraft vs MNK games

Our research plan was largely agnostic to the domain used. Here, we discuss strengths and weaknesses of two options. Ultimately, both are spatial and sequential domains, which are the main properties we sought. We considered using both StarCraft and MNK games, and chose to use the MNK domain, primarily because it offers greater scientific control.

Weaknesses of StarCraft: RTS games like StarCraft incur challenges for empirical study, the largest of which is that a game takes quite a long time (15 minutes up to an hour). Given the importance of exposing participants to many agents, we would need to find some ways to shorten tasks. One example could be to operate on a restriction of the game, such as a "micro battle" (each player starts with a set of units, cannot build more, and the last player standing wins). While we could create such a custom game, StarCraft does not offer as much control over the software stack as a game of our own creation.

Strengths of StarCraft: The main strengths of using StarCraft are that it is flashy and would synchronize better with the rest of the XAI project. Additionally, we have found that there is no shortage of StarCraft enthusiasts at Oregon State, so recruitment should be no issue. Also, we have already invested significant time and effort studying people performing tasks in the StarCraft domain in Steps 1 and 2.

Weaknesses of MNK: One of the weaknesses of the MNK domain lies in its simplicity. Indeed, we have discussed applying deep learning to a domain that can be solved well by a number of other strategies (e.g. value iteration or search). However, there are many domains where these techniques do not work. Further, it may prove challenging to devise explanation strategies that work across multiple types of learner.

Strengths of MNK: The main strengths of the MNK domain arise from the same source as its weakness—simplicity. In MNK games, each player attempts to make a sequence of length K on a board of size M $\times$ N—most people are familiar with Tic-Tac-Toe, which would be 3-3-3. Each square can be represented with a vector of 2-3 bits—opponent controlled, friendly controlled, empty (optional). This binary representation

works because either a piece is there or it is not. Thus, generating various types of counterfactual explanations (such as the Sensitivity-based explanation in [46]) becomes simpler because the alternative options are well defined and bounded in number. Further, the move tree has a bounded depth—eventually the board will be full. This allows for the use of random rollouts to estimate the quality of non-terminal states, which is a property utilized by AlphaGo [164]. Note that the Go domain has a similar representation as MNK games, and could be considered as a domain for study. However, unlike Go, most people are familiar with Tic-Tac-Toe, making tutorializing MNK simple. Assisting tutorializing is the fact that in this domain, *"people's intuitive priors (three-in-a-row is good) happen to be correct"* [188]. Indeed, composing these priors into effective heuristics is an ongoing research topic in MNK games (e.g. [4]).

MNK games have some other nice properties, such as variation of known optimal behavior. In certain configurations, optimal play from both sides will result in a draw; for example Tic-Tac-Toe falls in this category[1] In other configurations, the player acting first can force a win [188]. Thus, it will be interesting to see if the agent discovers known strategies when they exist.

Another advantage of using the MNK games domain is that despite the simplicity of the mechanics of MNK games, the general population does not hold a deep knowledge of the domain. In our studies, we have found games with widely varying experience levels to increase the challenge of separating predictions based on a participants *domain knowledge* from those based on their *mental model of the agent.* In this sense, MNK offers a domain which is both unfamiliar and understandable, which might help us separate predictions based on domain knowledge from those based on the mental model of the AI's behavior. In fact, work on human sequential-decision making has illustrated two means to evaluate participant domain knowledge via a series of "two-alternative forced choice" options (ground truthed by comparing responses with decisions from an optimal search-based agent) or a series of evaluations of winning chance given a board state on a Likert scale (ground truthed by correlating responses with an optimal agent's rankings) [188].

Another set of important properties of the MNK domain that will assist training agents with varying strengths is that it has a known transition model. This means we can

[1] Uiterwijk [184] recently proved that on our 4x9 board, the player going first *wins*, given optimal play on both sides. Hsieh and Tsai [73] offer some additional results of this kind, as well as reductions demonstrating the complexity of various configurations.

create model-based agents without high time and compute costs associated with learning the transition function. Similarly, MNK offers rotational and reflectional symmetries which provides another way to accelerate the learning process [135, 152, 151]. Notably, both of these properties can be used to train agents of varying strengths (i.e. comparing model-based vs model-free agents or agents that experience symmetric states vs ones that do not).

Additionally, we can vary the difficulty of the task, *both* in terms of participant foraging difficulty *and* in terms of strategic complexity. To do so, we simply adjusting the parameters of M, N, and K. In contrast, we have found full StarCraft episodes to be a consistently difficult foraging challenge [136]. To clarify, participants varied greatly in terms of how they performed camera controls. This led them to observe different actions during the same time slice—a very detrimental source of variation for controlled studies of the impact of explanation.

Last, MNK games proceed very quickly, so observing many agents will be no trouble. While the game may be slightly foreign to some participants, it has simple mechanics and can be quickly tutorialized. StarCraft games proceed slowly, and teaching participants the particulars of a restricted version of StarCraft will take some time and may prove confusing to experienced players. To clarify, when tutorializing a restriction of StarCraft, we will need to be very careful and clear about what parts of the game *are* StarCraft, and which ones *are not*.

Appendix E: Log Appendix - Making a move with our CNNAgent

```
Player 1-MiniNoiseLayer5 preparing to make a move, playing X on this board

---X-OOO-
----X-X--
--XOX-XX-
---O--OOO

NN outcome tensor (transposed to align with the sampled tensor)
tensor([[[    0.3196,     0.6669,     0.0000],
         [    0.2872,     0.6986,     0.0000],
         [    0.2624,     0.7332,     0.0000],
         [    0.3336,     0.6600,     0.0000]],

        [[    0.2688,     0.7223,     0.0000],
         [    0.2865,     0.6934,     0.0000],
         [    0.3542,     0.6413,     0.0000],
         [    0.3083,     0.6810,     0.0000]],

        [[    0.3754,     0.6148,     0.0000],
         [    0.4165,     0.5647,     0.0000],
         [    0.0330,     0.9660,     0.0000],
         [    0.3313,     0.6515,     0.0000]],

        [[    0.4672,     0.5246,     0.0000],
         [    0.4339,     0.5633,     0.0000],
         [    0.0000,     1.0000,     0.0000],
         [    0.1230,     0.8766,     0.0000]],

        [[    0.2373,     0.7574,     0.0000],
         [    0.0547,     0.9432,     0.0000],
         [    0.0480,     0.9486,     0.0000],
         [    0.1768,     0.8170,     0.0000]],

        [[    0.2803,     0.7106,     0.0000],
         [    0.3575,     0.6249,     0.0000],
```

```
         [    0.6181,     0.3773,     0.0000],
         [    0.2191,     0.7697,     0.0000]],

        [[    0.0436,     0.9575,     0.0000],
         [    0.0000,     1.0000,     0.0000],
         [    0.0000,     1.0000,     0.0000],
         [    0.0167,     0.9828,     0.0000]],

        [[    0.2090,     0.7908,     0.0000],
         [    0.2970,     0.6972,     0.0000],
         [    0.0000,     1.0000,     0.0000],
         [    0.0251,     0.9758,     0.0000]],

        [[    0.1845,     0.8084,     0.0000],
         [    0.2708,     0.7241,     0.0000],
         [    0.3259,     0.6641,     0.0000],
         [    0.1328,     0.8639,     0.0000]]], grad_fn=<TransposeBackward0>)

Score Matrix
tensor([[-0.3474, -0.4534, -0.2394, -0.0575, -0.5202, -0.4304, -0.9139, -0.5818, -0.6239],
        [-0.4113, -0.4069, -0.1482, -0.1294, -0.8885, -0.2674, -1.0000, -0.4002, -0.4533],
        [-0.4708, -0.2871, -0.9330, -1.0000, -0.9006,  0.2408, -1.0000, -1.0000, -0.3382],
        [-0.3264, -0.3727, -0.3202, -0.7536, -0.6402, -0.5505, -0.9661, -0.9507, -0.7311]],
        grad_fn=<AddBackward0>)

Post-CONSTRAINT Score Matrix
tensor([[-0.3474, -0.4534, -0.2394, -1.0000, -0.5202, -1.0000, -1.0000, -1.0000, -0.6239],
        [-0.4113, -0.4069, -0.1482, -0.1294, -1.0000, -0.2674, -1.0000, -0.4002, -0.4533],
        [-0.4708, -0.2871, -1.0000, -1.0000, -1.0000,  0.2408, -1.0000, -1.0000, -0.3382],
        [-0.3264, -0.3727, -0.3202, -1.0000, -0.6402, -0.5505, -1.0000, -1.0000, -1.0000]],
        grad_fn=<CopySlices>)

Temperature: 0.1
ActionProbs
tensor([[0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0.]], grad_fn=<ViewBackward>)
Move: 5 2
```

```
Final Board:
---X-OOO-
----X-X--
--XOXXXX-
---O--OOO

MoveLog: [(3, 3), (6, 1), (7, 3), (6, 2), (7, 0), (7, 2), (6, 0),
(2, 2), (3, 2), (3, 0),(5, 0), (4, 1), (8, 3), (4, 2), (6, 3), (5, 2)]
```

Appendix F: Log Appendix - Computing a target tensor with CNN Agent

```
--XOXXXO-
OO--X----
--O--O-O-
XX-X--O--



TARGET outcome tensor (from sampling 10 games per move):
tensor([[[0.0000, 0.5000, 0.5000],
         [0.0000, 1.0000, 0.0000],
         [0.0000, 0.7000, 0.3000],
         [0.0000, 1.0000, 0.0000]],

        [[0.0000, 0.6000, 0.4000],
         [0.0000, 1.0000, 0.0000],
         [0.0000, 0.5000, 0.5000],
         [0.0000, 1.0000, 0.0000]],

        [[0.0000, 1.0000, 0.0000],
         [0.0000, 0.3000, 0.7000],
         [0.0000, 1.0000, 0.0000],
         [0.0000, 0.0000, 1.0000]],

        [[0.0000, 1.0000, 0.0000],
         [0.0000, 0.5000, 0.5000],
         [0.0000, 0.2000, 0.8000],
         [0.0000, 1.0000, 0.0000]],

        [[0.0000, 1.0000, 0.0000],
         [0.0000, 1.0000, 0.0000],
         [0.0000, 0.2000, 0.8000],
         [0.0000, 0.5000, 0.5000]],
```

```
         [[0.0000, 1.0000, 0.0000],
          [0.0000, 0.4000, 0.6000],
          [0.0000, 1.0000, 0.0000],
          [0.0000, 0.5000, 0.5000]],

         [[0.0000, 1.0000, 0.0000],
          [0.0000, 0.7000, 0.3000],
          [0.0000, 0.3000, 0.7000],
          [0.0000, 1.0000, 0.0000]],

         [[0.0000, 1.0000, 0.0000],
          [0.0000, 0.6000, 0.4000],
          [0.0000, 1.0000, 0.0000],
          [0.0000, 0.6000, 0.4000]],

         [[0.0000, 0.3000, 0.7000],
          [0.0000, 0.8000, 0.2000],
          [0.0000, 0.4000, 0.6000],
          [0.0000, 0.8000, 0.2000]]])
OUTPUT outcome tensor (from NN):
tensor([[[ 0.0738,  0.6205,  0.2897],
         [ 0.0189,  0.7158,  0.1611],
         [ 0.0566,  0.5442,  0.3460],
         [ 0.0731,  0.5125,  0.3862]],

         [[-0.0382,  0.5769,  0.3813],
          [ 0.0109,  0.6496,  0.3024],
          [-0.0154,  0.6286,  0.3214],
          [-0.0119,  0.8577,  0.0228]],

         [[ 0.0222,  0.7372,  0.2692],
          [-0.0204,  0.6188,  0.3473],
          [ 0.0338,  0.7412,  0.2488],
          [ 0.0510,  0.5191,  0.4941]],

         [[ 0.0108,  0.7104,  0.2483],
          [-0.0432,  0.5495,  0.3483],
          [-0.0162,  0.5925,  0.3710],
          [ 0.0108,  0.6629,  0.2960]],
```

```
[[ 0.0131,  0.6355,  0.3294],
 [-0.0114,  0.7674,  0.1904],
 [-0.0295,  0.5565,  0.4217],
 [ 0.0157,  0.5400,  0.3936]],

[[ 0.0155,  0.6555,  0.2742],
 [-0.0099,  0.5135,  0.3988],
 [ 0.0269,  0.6449,  0.3474],
 [ 0.0191,  0.6692,  0.2868]],

[[-0.0293,  0.6932,  0.2932],
 [ 0.0155,  0.6482,  0.3496],
 [ 0.0284,  0.7241,  0.2228],
 [-0.0012,  0.7389,  0.2315]],

[[-0.0258,  0.5164,  0.4822],
 [-0.0120,  0.6888,  0.2677],
 [-0.0039,  0.7739,  0.1482],
 [-0.0123,  0.6237,  0.2988]],

[[ 0.0129,  0.7177,  0.2733],
 [ 0.0114,  0.6529,  0.2722],
 [ 0.0154,  0.5309,  0.4107],
 [-0.0030,  0.7000,  0.2979]]], grad_fn=<TransposeBackward0>)
```

Appendix G: How to manipulate the agent "quality"?

We will manipulate the underlying quality of the agents using the following methods:

1. **Sampling** a wide variety of agent configurations (say at various points along the learning curve of a single training session) and/or
2. **Training** a set of agents with different hyperparameter settings or with differing amounts of experience based on whether the training process uses symmetries or not.
3. **Poisoning** the training data (e.g. corrupting rewards during the training process)
4. **Mutating** a single trained agent (e.g. modifying individual weights after training).

The idea with method #1 is that it is often easy (in terms of human labor, not necessarily compute time) for an AI's creators to generate many system configurations. In fact, it may be a necessary part of the training process to create many of these configurations, as shown in Figure G.1. Also, this strategy has already been used in an A vs B testing by Huang et al. [74] in their study of which states to show an assessor in a simulated car driving domain. They found showing "critical" states, as defined by their function, led participants to rate the more trained agent higher than the less trained one.

Further, final configuration depends on more than just training time, but includes training *process*. A good example of this is choosing hyperparameter values, often done via grid or line search ([56] Chapter 11), meaning many configurations are searched to empirically choose the best performing. Another possibly useful way to vary the training process is to use different architectures or processes which do (or do not) use symmetries inherent to the MNK domain to accelerate learning, specifically rotational and/or reflectional symmetry. In effect, we might expect an agent that must learn of any symmetries present in the domain will be outperformed by those for which those symmetries are explicitly represented. This idea is similar to a user study by Rosenfeld et al. [152] asking participants to devise similarity functions (e.g. symmetry) to increase the performance of the system.

In order to create systematic issues, one of the most straightforward approaches is to poison the training data. To illustrate, consider that the simulator managing the game can tell the agent it won or lost at will. So if we wish to discourage certain types of actions in certain types of situations, it may be possible to accomplish this by programming the simulator to tell the agent it won when it actually lost (or vice versa) in certain situations.

The idea with the fourth method is to draw inspiration from software testing literature to generate and kill mutants. This technique was first published in 1978 [38], but is still used today [137]. In mutant testing, the first step is to generate mutants by manipulating the source code many times (e.g. replacing a "+" with a "-"), each time creating a different mutant. Then, the quality of a testing methodology can be measured by determining how many of the mutants in the set are detected and "killed." Thus, we can measure the efficacy of the person-machine team of human + explanation system in a similar way—our "testing suite."

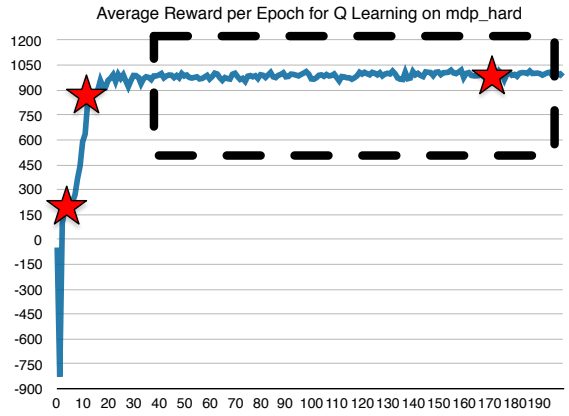


Figure G.1: For illustrative purposes, an example learning curve, reward gained per training time. First, note that a number of these system configurations are not the same, but perform similarly in the aggregate (highlighted with the dashed black box). Which one should the developer submit for final approval? Second, note that this figure illustrates how a sampling approach can be used to obtain agents of varying quality. Specifically, choosing agents with configurations marked with the red ★ will yield "good," "bad," and "medium" agents.

Applying a strategy inspired by mutant generation in our framework will allow us to measure how well *explanations* allow participants to detect bugs that we have intentionally introduced. In fact, a technique similar to mutant generation has already been employed to investigate whether saliency-based explanations are based in the data or the model [6]. In that work, the authors randomize a portion of the a deep network and examine how the saliency maps generated by that model change (essentially comparing *rows* of Figure G.2). Then, they argue that if drastic changes to the model result in only small changes to the saliency map, then the saliency technique is based on the *data* only, and not the *model*. Thus, using such a saliency technique would not be helpful to assess the model (e.g., Guided back-propagation or Guided GradCAM).

Our observation is that we can use this approach to generate a set of agents which are "damaged" to varying degrees, given an initial well-trained agent. To illustrate, compare the *columns* of Figure G.2—randomizing weights on layers closer to the input effected larger changes on the saliency explanation generated.

One of the disadvantages of mutant generation is that it is the least ecologically valid of the manipulations. Some of the manipulation strategies have a simple cause that will cause systematic errors (e.g. the agent is blind to a square or unable to use a square via poisoning). However, it is unclear how mutating weights will affect action selection, in that incorrect decisions may not appear in any kind of systematic way.

Using all of these techniques, we can create a set of agents for the agents to assess like the following

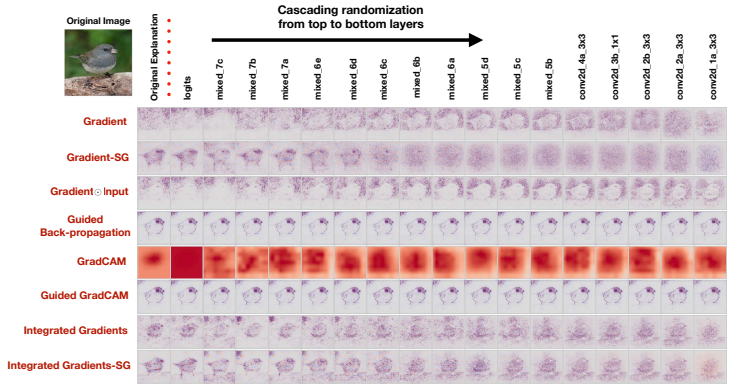


Figure G.2: This is Figure 9 from Adebayo et al. [6]. In this figure, they are *randomizing* the weights on a single layer of a deep network doing image classification. The layer being randomized varies from nearest the output (`logits`) to nearest input (`conv2d_1a_3x3`). In each row, they show the results from different saliency techniques under such modification. They use this figure to argue that certain techniques are inappropriate for use to inspect the model, as their output does not vary under this significant change to the model (e.g. `Guided Back-propagation` and `Guided GradCAM`). However, interpreting the figure columnwise, the amount of "damage" to the network varies fairly smoothly as the randomization moves between layers. This indicates that randomizing weights of particular layers could provide agents of varying quality.

Agent Name	Record vs each other	Record vs Heuristic Agents
vsMainPolicy1k	[1000, 0, 0]	[292, 291, 17]
sample0500	[300, 700, 0]	[47, 529, 24]
sample1000	[400, 600, 0]	[53, 520, 27]
sample1500	[400, 600, 0]	[53, 539, 8]
sample2000	[400, 600, 0]	[64, 518, 18]
sample2500	[450, 550, 0]	[55, 533, 12]
sample3000	[150, 850, 0]	[54, 502, 44]
sample3500	[550, 450, 0]	[70, 495, 35]
sample4000	[700, 300, 0]	[79, 484, 37]
sample4500	[650, 350, 0]	[75, 490, 35]
sample5000	[500, 500, 0]	[72, 496, 32]

Table G.1: Performance benchmarks associated with each agent created by *sampling*, using random rollouts. Here, "each other" refers to the listed agents, which includes "vsMainPolicy1k" as benchmark against the best NN trained so far. The "heuristic agents" are the same in Tables G.1–G.4 (HeurBaseK, HeurBase3, HeurBase2, Aggressive, Defensive, and bRANDy). See Appendix K for full data.

nine: {`best, sample1, sample2, training1, training2, poison1, poison2, mutant1, mutant2`}. Note that in this formulation, `best` would be the base agent, while we manipulate the rest with respect to a key property—quality. In essence, the quality of each manipulated agent will be diminished to varying degrees. Then, it will be up to participants to differentiate amongst the group, using explanations.

G.1 Verdicts

1. **Sampling (Verdict: Fair)** - The idea here is to dump out model configurations periodically as training progresses; here we dump a weight file each time $\frac{1}{10}$th of the training duration is completed. Table G.1 shows the results from a short training session, and shows how performance tends to increase with more experience. This is as anticipated, but this approach has been used in prior work by Huang et al. [74].
2. **Training (Verdict: Good)** - In order to understand well the pipeline worked when encouraging overfitting to the heuristic functions described earlier, I started by training up six neural networks. Three were trained using random rollouts, and three trained using policy rollouts, which increase training time substantially. None of these CNN agents surpassed the best of the heuristic functions, but they did show a smooth gradation of quality, demonstrated in Table G.2.
3. **Poisoning (Verdict: Bad)** - This is implemented by exporting the board to the tensor, and then modifying it in the following concrete ways:
   (a) Broken Sensor - At construction time, an increasing number of squares are be selected with a seeded random selection process to be misperceived. By modifying the *input* to the neural network, the agent sees these squares as all containing either (1) an *opposing* piece (2) a

Agent Name	Training Time	Record vs each other	Record vs Heuristic Agents
vsMainPolicy1k	6.5 hrs	[450, 50, 0]	[293, 288, 19]
vsRandom10k	4.0 hrs	[450, 50, 0]	[122, 445, 33]
vsAggro10k	3.0 hrs	[300, 200, 0]	[131, 446, 23]
vsMain10k	3.5 hrs	[150, 350, 0]	[93, 472, 35]
vsDefensive10k	7.5 hrs	[150, 350, 0]	[56, 474, 70]
vsDefPolicy1k	86.0 hrs	[0, 500, 0]	[34, 505, 61]

Table G.2: Performance benchmarks associated with each agent created by *training* to encourage overfitting, by *only* playing against a specific heuristic function. Training times are rounded to the nearest half-hour, and records are presented as [W, L D]. Tables G.1–G.4 all will use a similar format, showing the record of games between the agents listed in the table ("each other") and between the agent in a row of the table vs a gauntlet of heuristic agents (HeurBaseK, HeurBase3, HeurBase2, Aggressive, Defensive, and Random). See Appendix L for full data.

friendly piece.

(b) Broken Actuator - After the neural network provides its output, the agent enforces domain constraints by zeroing out probabilities for illegal moves. By "erroneously" modifying the *output* of the neural network, the agent is able to correctly *perceive* the board, but *unwilling* to make certain moves if the probabilities are zeroed (or *compelled* to make certain moves if the probabilities are maxed).

Poisoning, when applied to perception, appears to not have the expected effect, behaving rather wildly. This could be that our agent is insufficiently converged for this kind of damage to be evident. However, changing how the actuators work in this way seems to damage the agent in increasing fashion as one might expect. It could be that *training* under poisoned circumstances provides better results than the current approach, which is to train normally, then apply blinders. However, spinning up new training processes can be expensive, and is undesirable for the purposes of testing.

4. **Mutating (Verdict: Excellent)** - This is implemented by adding noise from the Gaussian distribution to a target layer. Here I always use a mean of 0 and let the standard deviation parameter vary in addition to the target layer parameter. This experiment acted exactly as anticipated by Figure G.2), in that noisifying layers close to the output seemed to debilitate the agent far more. This approach appears to be the most highly controllable way to generate agents of differing quality, and also happens to not require wasting a great deal of training effort duplication by training vs specific opponents.

Agent Name	Record vs each other	Record vs Heuristic Agents
vsMainPolicy1k	[750, 450, 0]	[289, 301, 10]
poiPercBehind1	[800, 400, 0]	[291, 294, 15]
poiPercBehind2	[750, 450, 0]	[282, 303, 15]
poiPercBehind5	[700, 500, 0]	[292, 290, 18]
poiPercAhead1	[750, 450, 0]	[287, 298, 15]
poiPercAhead2	[850, 350, 0]	[291, 290, 19]
poiPercAhead5	[600, 600, 0]	[291, 295, 14]
poiCompulsiveAct1	[750, 450, 0]	[289, 297, 14]
poiCompulsiveAct2	[1100, 100, 0]	[106, 480, 14]
poiCompulsiveAct5	[300, 900, 0]	[70, 527, 3]
poiTabooAct1	[150, 1050, 0]	[55, 524, 21]
poiTabooAct2	[150, 1050, 0]	[58, 525, 17]
poiTabooAct5	[150, 1050, 0]	[44, 542, 14]

Table G.3: Performance benchmarks associated with each agent created by poisoning. The "heuristic agents" are the same in Tables G.1–G.4 (HeurBaseK, HeurBase3, HeurBase2, Aggressive, Defensive, and bRANDy). There are 4 kinds of poisoning shown here, each applied to 1, 2, and 5 squares. In the first block, the poisoned squares are seen as controlled by the *opponent*—regardless of their state, while in the second block, the poisoned squares are seen as controlled by the *player*. In the third block, the poisoned squares receive maximum probability, so those moves must be made before any others can be considered. Meanwhile, in the fourth block, the poisoned squares receive minimum probability, so the agent will always be unwilling to make those moves. See Appendix M for full data.

Agent Name	Noise SD	Record vs each other	Record vs Heuristic Agents
vsMainPolicy1k	-	[600, 600, 0]	[290, 294, 16]
mutConv1SD100	100	[450, 700, 50]	[144, 447, 9]
mutConv1SD10	10	[1050, 150, 0]	[134, 456, 10]
mutConv1SD1	1	[850, 350, 0]	[288, 289, 23]
mutConv1SDp1	.1	[500, 700, 0]	[291, 301, 8]
mutConv2SD100	100	[150, 1050, 0]	[43, 552, 5]
mutConv2SD10	10	[900, 300, 0]	[68, 513, 19]
mutConv2SD1	1	[600, 600, 0]	[232, 353, 15]
mutConv2SDp1	.1	[700, 450, 50]	[290, 286, 24]
mutFc1SD100	100	[0, 1100, 100]	[0, 576, 24]
mutFc1SD1	1	[450, 750, 0]	[47, 540, 13]
mutFc1SDp1	.1	[650, 450, 100]	[146, 440, 14]
mutFc1SDp01	.01	[700, 400, 100]	[288, 302, 10]

Table G.4: Performance benchmarks associated with each agent created by mutation. Starting from a baseline of the `vsMainPolicy1k` agent, one of the 3 layers (input $\rightarrow$ conv1 $\rightarrow$ conv2 $\rightarrow$ fc1 $\rightarrow$ output) is perturbed by adding Gaussian noise with $mean = 0$ and SD as specified. Note that the layers near the output seem to become increasingly sensitive to weight noisification, and that strengthening the noise has an increasingly deleterious affect on the agent at *all* layers. The "heuristic agents" are the same in Tables G.1–G.4 (HeurBaseK, HeurBase3, HeurBase2, Aggressive, Defensive, and bRANDy). See Appendix N for full data.

Appendix H: The Heuristic Agent

To define the state evaluation function, I started with the following from Russel Norvig[154], for the Tic-Tac-Toe domain (also known as 3,3,3, since my program takes M,N, and K as arguments to the `Frame` constructor).

$$Eval_{Tic-Tac-Toe}(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s)) \tag{H.1}$$

Following the heuristic in Equation H.1 seemed to result in a good policy for 3,3,3, so I formed a first design goal: to have generalizations of this heuristic I created for other MNK to result in this one if evaluated with those values. The second design goal was to use a simpler[1] expression than Abdoulaye et al. [4]. To that end, I derived several candidates, which generalize the idea of counting the number of possible winning "arrays" which are both uncontested and contain a specific number of pieces. In the presence of ties for the maximal heuristic evaluation, the agent randomly chooses a move from those involved in the tie.

$$Eval_2(s) = \sum_{i=1}^{k-1}(2^i - 1)(X_i(s) - O_i(s)) \tag{H.2}$$

$$Eval_3(s) = \sum_{i=1}^{k-1}3^{i-1}(X_i(s) - O_i(s)) \tag{H.3}$$

$$Eval_k(s) = \sum_{i=1}^{k-1}k^{i-1}(X_i(s) - O_i(s)) \tag{H.4}$$

To pick amongst these, I tested them empirically for several values of MNK (results shown in Appendix J). While $Eval_k$ (Equation H.4) is comparable to the best, though I thought of it after having done the rest of the experiments presented in this update, which uses $Eval_3$.

With a heuristic function in hand, I realized I could meet another design goal from the prelim—specifically, creating "situations" which are meaningfully distinct in a strategic sense. Note that each heuristic function is composed of two kinds of terms, positive ones for friendly pieces and negative ones for enemy pieces. So, from the heuristic function, we create 3 agents, one *balanced* (following Equation H.4), one *aggressive* (following Equation H.5), and one *defensive* (following Equation H.6). By training a CNN agent against specific opponents from this set, it is possible to encourage overfitting to the opponent,

[1] The expression they derive looks like this:

$$A = \begin{cases} \sum_{i=1}^{k-3}(a_{2i} - 1p_{i,1} + a_{2i}p_{i,2}) + a_{2(k-2)-1}p_{k-2,1} + 100p_{k-2,2} + 80p_{k-1,1} + 250p_{k-1,2} + 10000000p_k & if k > 3 \\ a_1p_{k-2,1} + 100p_{k-2,2} + 80p_{k-1,1} + 250p_{k-1,2} + 10000000p_k & if k = 3 \end{cases}$$

resulting in the meaningful distinctions between the resultant CNN agents. See Appendix I for a log generated by making a move with a heuristic agent.

$$Eval_k(s) = \sum_{i=1}^{k-1} k^{i-1}(X_i(s)) \tag{H.5}$$

$$Eval_k(s) = \sum_{i=1}^{k-1} k^{i-1}(-O_i(s)) \tag{H.6}$$

Appendix I: Log Appendix - Moving with Heuristic Agents

```
Player HeurBase3 is preparing to make a move, playing O

---OXXOX-
---XX----
---O-----
--X-OO---
*** Heuristic fn scoring complete, results:
square SCORE good bad coeffs
(0, 0) 4 [11, 4, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(1, 0) 9 [10, 3, 1, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(2, 0) 3 [9, 4, 0, 0] [6, 4, 0, 0] [1, 3, 9, 27]
(8, 0) 3 [10, 4, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(0, 1) 2 [11, 3, 0, 0] [6, 4, 0, 0] [1, 3, 9, 27]
(1, 1) 6 [12, 3, 0, 0] [6, 3, 0, 0] [1, 3, 9, 27]
(2, 1) 16 [9, 3, 1, 0] [5, 2, 0, 0] [1, 3, 9, 27]
(5, 1) 11 [10, 4, 0, 0] [5, 2, 0, 0] [1, 3, 9, 27]
(6, 1) 8 [10, 4, 0, 0] [5, 3, 0, 0] [1, 3, 9, 27]
(7, 1) 5 [10, 4, 0, 0] [5, 4, 0, 0] [1, 3, 9, 27]
(8, 1) 2 [12, 3, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(0, 2) 3 [10, 4, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(1, 2) 7 [8, 6, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(2, 2) 11 [8, 6, 0, 0] [6, 3, 0, 0] [1, 3, 9, 27]
(4, 2) 9 [7, 7, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(5, 2) 6 [10, 5, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(6, 2) 9 [9, 6, 0, 0] [6, 4, 0, 0] [1, 3, 9, 27]
(7, 2) 4 [12, 3, 0, 0] [5, 4, 0, 0] [1, 3, 9, 27]
(8, 2) 2 [12, 3, 0, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(0, 3) 4 [10, 4, 0, 0] [6, 4, 0, 0] [1, 3, 9, 27]
(1, 3) 5 [11, 3, 0, 0] [6, 3, 0, 0] [1, 3, 9, 27]
(3, 3) 10 [10, 3, 1, 0] [6, 4, 0, 0] [1, 3, 9, 27]
(6, 3) 16 [8, 3, 2, 0] [7, 4, 0, 0] [1, 3, 9, 27]
(7, 3) 10 [9, 3, 1, 0] [5, 4, 0, 0] [1, 3, 9, 27]
(8, 3) 4 [10, 4, 0, 0] [6, 4, 0, 0] [1, 3, 9, 27]

Best Options: [ 6 22]
```

```
 Final Answer: (6, 3)
-----------------------------------
Player HeurBaseK is preparing to make a move, playing X

---OXXOX-
---XX----
---O-----
--X-OOO--
*** Heuristic fn scoring complete, results:
square SCORE good bad coeffs
(0, 0) -26 [9, 4, 0, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(1, 0) -23 [8, 4, 0, 0] [7, 2, 2, 0] [1, 4, 16, 64]
(2, 0) -25 [6, 5, 0, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(8, 0) -27 [8, 4, 0, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(0, 1) -25 [7, 5, 0, 0] [8, 3, 2, 0] [1, 4, 16, 64]
(1, 1) -12 [8, 4, 1, 0] [8, 3, 2, 0] [1, 4, 16, 64]
(2, 1) 6 [5, 4, 2, 0] [7, 2, 2, 0] [1, 4, 16, 64]
(5, 1) 3 [6, 4, 2, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(6, 1) -6 [6, 5, 1, 0] [8, 2, 2, 0] [1, 4, 16, 64]
(7, 1) -21 [6, 6, 0, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(8, 1) -27 [9, 4, 0, 0] [8, 3, 2, 0] [1, 4, 16, 64]
(0, 2) -27 [8, 4, 0, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(1, 2) -25 [8, 4, 0, 0] [5, 3, 2, 0] [1, 4, 16, 64]
(2, 2) -10 [7, 4, 1, 0] [5, 3, 2, 0] [1, 4, 16, 64]
(4, 2) -24 [8, 4, 0, 0] [4, 3, 2, 0] [1, 4, 16, 64]
(5, 2) -25 [9, 4, 0, 0] [6, 3, 2, 0] [1, 4, 16, 64]
(6, 2) -18 [8, 5, 0, 0] [6, 2, 2, 0] [1, 4, 16, 64]
(7, 2) -21 [7, 6, 0, 0] [8, 3, 2, 0] [1, 4, 16, 64]
(8, 2) -27 [9, 4, 0, 0] [8, 3, 2, 0] [1, 4, 16, 64]
(0, 3) -24 [7, 5, 0, 0] [7, 3, 2, 0] [1, 4, 16, 64]
(1, 3) -13 [7, 4, 1, 0] [8, 3, 2, 0] [1, 4, 16, 64]
(3, 3) -8 [7, 5, 0, 0] [7, 3, 1, 0] [1, 4, 16, 64]
(7, 3) -3 [5, 6, 0, 0] [8, 2, 1, 0] [1, 4, 16, 64]
(8, 3) -21 [7, 5, 0, 0] [8, 2, 2, 0] [1, 4, 16, 64]

Best Options: [6]
 Final Answer: (2, 1)
```

Appendix J: Data Appendix - Choosing a Heuristic Function

J.1 For 3,3,3

	HeurBaseK	HeurBase3	HeurBase2	Aggressive	Defensive	bRANDy	TOTAL (W, L , D)
HeurBaseK	- - - - -	[0, 0, 100]	[0, 0, 100]	[100, 0, 0]	[33, 1, 66]	[92, 1, 7]	[225, 2, 273]
HeurBase3	[0, 0, 100]	- - - - -	[0, 0, 100]	[100, 0, 0]	[24, 1, 75]	[93, 0, 7]	[217, 1, 282]
HeurBase2	[0, 0, 100]	[0, 0, 100]	- - - - -	[100, 0, 0]	[27, 1, 72]	[95, 0, 5]	[222, 1, 277]
Aggressive	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[30, 70, 0]	[93, 7, 0]	[123, 377, 0]
Defensive	[1, 33, 66]	[1, 24, 75]	[1, 27, 72]	[70, 30, 0]	- - - - -	[58, 1, 41]	[131, 115, 254]
bRANDy	[1, 92, 7]	[0, 93, 7]	[0, 95, 5]	[7, 93, 0]	[1, 58, 41]	- - - - -	[9, 431, 60]

J.2 For 9,4,4

	HeurBaseK	HeurBase3	HeurBase2	Aggressive	Defensive	bRANDy	TOTAL (W, L , D)
HeurBaseK	- - - - -	[51, 47, 2]	[66, 31, 3]	[90, 10, 0]	[23, 0, 77]	[100, 0, 0]	[330, 88, 82]
HeurBase3	[47, 51, 2]	- - - - -	[71, 26, 3]	[93, 7, 0]	[25, 0, 75]	[100, 0, 0]	[336, 84, 80]
HeurBase2	[31, 66, 3]	[26, 71, 3]	- - - - -	[64, 36, 0]	[18, 1, 81]	[100, 0, 0]	[239, 174, 87]
Aggressive	[10, 90, 0]	[7, 93, 0]	[36, 64, 0]	- - - - -	[38, 62, 0]	[100, 0, 0]	[191, 309, 0]
Defensive	[0, 23, 77]	[0, 25, 75]	[1, 18, 81]	[62, 38, 0]	- - - - -	[80, 0, 20]	[143, 104, 253]
bRANDy	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 80, 20]	- - - - -	[0, 480, 20]

J.3 For 12,7,5

	HeurBaseK	HeurBase3	HeurBase2	Aggressive	Defensive	bRANDy	TOTAL (W, L , D)
HeurBaseK	- - - - -	[47, 53, 0]	[90, 10, 0]	[71, 29, 0]	[6, 0, 94]	[100, 0, 0]	[314, 92, 94]
HeurBase3	[53, 47, 0]	- - - - -	[80, 20, 0]	[81, 19, 0]	[22, 0, 78]	[100, 0, 0]	[336, 86, 78]
HeurBase2	[10, 90, 0]	[20, 80, 0]	- - - - -	[23, 77, 0]	[15, 0, 85]	[100, 0, 0]	[168, 247, 85]
Aggressive	[29, 71, 0]	[19, 81, 0]	[77, 23, 0]	- - - - -	[17, 83, 0]	[100, 0, 0]	[242, 258, 0]
Defensive	[0, 6, 94]	[0, 22, 78]	[0, 15, 85]	[83, 17, 0]	- - - - -	[89, 0, 11]	[172, 60, 268]
bRANDy	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 89, 11]	- - - - -	[0, 489, 11]

Appendix K: Data Appendix - CNNs sampled throughout a training cycle

K.1 Sampled CNNs, vs each other

	vsMainPolicy1k	sample0500	sample1000	sample1500	sample2000	sample2500	sample3000	sample3500	sample4000	sample4500	sample5000	TOTAL (1
vsMainPolicy1k	- - - - -	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[1000, 0, 0]
sample0500	[0, 100, 0]	- - - - -	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[300, 700, 0]
sample1000	[0, 100, 0]	[100, 0, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[400, 600, 0]
sample1500	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[400, 600, 0]
sample2000	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[100, 0, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[400, 600, 0]
sample2500	[0, 100, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[100, 0, 0]	[450, 550, 0]
sample3000	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[50, 50, 0]	- - - - -	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[150, 850, 0]
sample3500	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	- - - - -	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[550, 450, 0]
sample4000	[0, 100, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[700, 300, 0]
sample4500	[0, 100, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[650, 350, 0]
sample5000	[0, 100, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[500, 500, 0]

K.2 Sampled CNNs, vs heuristic agents

```
vsMainPolicy1k vs HeurBaseK: [50, 50, 0]
vsMainPolicy1k vs HeurBase3: [50, 50, 0]
vsMainPolicy1k vs HeurBase2: [50, 50, 0]
vsMainPolicy1k vs Aggressive: [50, 50, 0]
vsMainPolicy1k vs Defensive: [0, 83, 17]
vsMainPolicy1k vs bRANDy: [92, 8, 0]
Total score for vsMainPolicy1k [292, 291, 17]

sample0500 vs HeurBaseK: [0, 100, 0]
sample0500 vs HeurBase3: [0, 100, 0]
sample0500 vs HeurBase2: [0, 100, 0]
sample0500 vs Aggressive: [0, 100, 0]
sample0500 vs Defensive: [0, 76, 24]
sample0500 vs bRANDy: [47, 53, 0]
```

```
Total score for sample0500 [47, 529, 24]

sample1000 vs HeurBaseK: [0, 100, 0]
sample1000 vs HeurBase3: [0, 100, 0]
sample1000 vs HeurBase2: [0, 100, 0]
sample1000 vs Aggressive: [0, 100, 0]
sample1000 vs Defensive: [0, 73, 27]
sample1000 vs bRANDy: [53, 47, 0]
Total score for sample1000 [53, 520, 27]

sample1500 vs HeurBaseK: [0, 100, 0]
sample1500 vs HeurBase3: [0, 100, 0]
sample1500 vs HeurBase2: [0, 100, 0]
sample1500 vs Aggressive: [0, 100, 0]
sample1500 vs Defensive: [0, 93, 7]
sample1500 vs bRANDy: [53, 46, 1]
Total score for sample1500 [53, 539, 8]

sample2000 vs HeurBaseK: [0, 100, 0]
sample2000 vs HeurBase3: [0, 100, 0]
sample2000 vs HeurBase2: [0, 100, 0]
sample2000 vs Aggressive: [0, 100, 0]
sample2000 vs Defensive: [0, 82, 18]
sample2000 vs bRANDy: [64, 36, 0]
Total score for sample2000 [64, 518, 18]

sample2500 vs HeurBaseK: [0, 100, 0]
sample2500 vs HeurBase3: [0, 100, 0]
sample2500 vs HeurBase2: [0, 100, 0]
sample2500 vs Aggressive: [0, 100, 0]
sample2500 vs Defensive: [0, 88, 12]
sample2500 vs bRANDy: [55, 45, 0]
Total score for sample2500 [55, 533, 12]

sample3000 vs HeurBaseK: [0, 100, 0]
sample3000 vs HeurBase3: [0, 100, 0]
sample3000 vs HeurBase2: [0, 100, 0]
sample3000 vs Aggressive: [0, 100, 0]
sample3000 vs Defensive: [0, 58, 42]
sample3000 vs bRANDy: [54, 44, 2]
```

```
Total score for sample3000 [54, 502, 44]

sample3500 vs HeurBaseK: [0, 100, 0]
sample3500 vs HeurBase3: [0, 100, 0]
sample3500 vs HeurBase2: [0, 100, 0]
sample3500 vs Aggressive: [0, 100, 0]
sample3500 vs Defensive: [0, 66, 34]
sample3500 vs bRANDy: [70, 29, 1]
Total score for sample3500 [70, 495, 35]

sample4000 vs HeurBaseK: [0, 100, 0]
sample4000 vs HeurBase3: [0, 100, 0]
sample4000 vs HeurBase2: [0, 100, 0]
sample4000 vs Aggressive: [0, 100, 0]
sample4000 vs Defensive: [0, 63, 37]
sample4000 vs bRANDy: [79, 21, 0]
Total score for sample4000 [79, 484, 37]

sample4500 vs HeurBaseK: [0, 100, 0]
sample4500 vs HeurBase3: [0, 100, 0]
sample4500 vs HeurBase2: [0, 100, 0]
sample4500 vs Aggressive: [0, 100, 0]
sample4500 vs Defensive: [0, 66, 34]
sample4500 vs bRANDy: [75, 24, 1]
Total score for sample4500 [75, 490, 35]

sample5000 vs HeurBaseK: [0, 100, 0]
sample5000 vs HeurBase3: [0, 100, 0]
sample5000 vs HeurBase2: [0, 100, 0]
sample5000 vs Aggressive: [0, 100, 0]
sample5000 vs Defensive: [0, 68, 32]
sample5000 vs bRANDy: [72, 28, 0]
Total score for sample5000 [72, 496, 32]
```

Appendix L: Data Appendix - CNNs trained to overfit to opponents

L.1 CNNs trained to overfit, vs each other

	vsMainPolicy1k	vsAggro10k	vsDef10k	vsDefPol1k	vsMain10k	vsRandom10k	TOTAL (W, L , D)
vsMainPolicy1k	- - - - -	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[450, 50, 0]
vsAggro10k	[0, 100, 0]	- - - - -	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[0, 100, 0]	[300, 200, 0]
vsDef10k	[0, 100, 0]	[0, 100, 0]	- - - - -	[100, 0, 0]	[50, 50, 0]	[0, 100, 0]	[150, 350, 0]
vsDefPol1k	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[0, 100, 0]	[0, 100, 0]	[0, 500, 0]
vsMain10k	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[100, 0, 0]	- - - - -	[0, 100, 0]	[150, 350, 0]
vsRandom10k	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	- - - - -	[450, 50, 0]

L.2 CNNs trained to overfit, vs Heuristic functions

```
vsMainPolicy1k vs HeurBaseK: [50, 50, 0]
vsMainPolicy1k vs HeurBase3: [50, 50, 0]
vsMainPolicy1k vs HeurBase2: [50, 50, 0]
vsMainPolicy1k vs Aggressive: [50, 50, 0]
vsMainPolicy1k vs Defensive: [0, 81, 19]
vsMainPolicy1k vs bRANDy: [93, 7, 0]
Total score for vsMainPolicy1k [293, 288, 19]

vsAggro10k vs HeurBaseK: [6, 94, 0]
vsAggro10k vs HeurBase3: [0, 100, 0]
vsAggro10k vs HeurBase2: [1, 99, 0]
vsAggro10k vs Aggressive: [30, 70, 0]
vsAggro10k vs Defensive: [0, 77, 23]
vsAggro10k vs bRANDy: [94, 6, 0]
Total score for vsAggro10k [131, 446, 23]

vsDef10k vs HeurBaseK: [0, 100, 0]
vsDef10k vs HeurBase3: [0, 100, 0]
vsDef10k vs HeurBase2: [0, 100, 0]
vsDef10k vs Aggressive: [0, 100, 0]
vsDef10k vs Defensive: [0, 31, 69]
vsDef10k vs bRANDy: [56, 43, 1]
Total score for vsDef10k [56, 474, 70]
```

```
vsDefPol1k vs HeurBaseK: [0, 100, 0]
vsDefPol1k vs HeurBase3: [0, 100, 0]
vsDefPol1k vs HeurBase2: [0, 100, 0]
vsDefPol1k vs Aggressive: [0, 100, 0]
vsDefPol1k vs Defensive: [0, 41, 59]
vsDefPol1k vs bRANDy: [34, 64, 2]
Total score for vsDefPol1k [34, 505, 61]

vsMain10k vs HeurBaseK: [0, 100, 0]
vsMain10k vs HeurBase3: [0, 100, 0]
vsMain10k vs HeurBase2: [0, 100, 0]
vsMain10k vs Aggressive: [0, 100, 0]
vsMain10k vs Defensive: [1, 64, 35]
vsMain10k vs bRANDy: [92, 8, 0]
Total score for vsMain10k [93, 472, 35]

vsRandom10k vs HeurBaseK: [0, 100, 0]
vsRandom10k vs HeurBase3: [0, 100, 0]
vsRandom10k vs HeurBase2: [2, 98, 0]
vsRandom10k vs Aggressive: [23, 77, 0]
vsRandom10k vs Defensive: [0, 67, 33]
vsRandom10k vs bRANDy: [97, 3, 0]
Total score for vsRandom10k [122, 445, 33]
```

Appendix M: Data Appendix - CNNs poisoned by perception/action

M.1 Poisoned CNNs, vs each other

	vsMainPolicy1k	poiPercBehind1	poiPercBehind2	poiPercBehind5	poiPercAhead1	poiPercAhead2	poiPercAhead5	poiCompulsiveAct1	poiCompulsiveAct2	poiCompulsiveAct5	poiTabooAct1	poiTabooAct2	poiTabooAct5	TOTAL (W, L , D)
vsMainPolicy1k	- - - - -	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[750, 450, 0]
poiPercBehind1	[50, 50, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[800, 400, 0]
poiPercBehind2	[50, 50, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[750, 450, 0]
poiPercBehind5	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[700, 500, 0]
poiPercAhead1	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[750, 450, 0]
poiPercAhead2	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	- - - - -	[100, 0, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[850, 350, 0]
poiPercAhead5	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[600, 600, 0]
poiCompulsiveAct1	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[750, 450, 0]
poiCompulsiveAct2	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	- - - - -	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[1100, 100, 0]
poiCompulsiveAct5	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[300, 900, 0]
poiTabooAct1	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[150, 1050, 0]
poiTabooAct2	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[150, 1050, 0]
poiTabooAct5	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[150, 1050, 0]

M.2 Poisoned CNNs, vs Heuristic Functions

```
vsMainPolicy1k vs HeurBaseK: [50, 50, 0]
vsMainPolicy1k vs HeurBase3: [50, 50, 0]
vsMainPolicy1k vs HeurBase2: [50, 50, 0]
vsMainPolicy1k vs Aggressive: [50, 50, 0]
vsMainPolicy1k vs Defensive: [0, 90, 10]
vsMainPolicy1k vs bRANDy: [89, 11, 0]
Total score for vsMainPolicy1k [289, 301, 10]

poiPercBehind1 vs HeurBaseK: [50, 50, 0]
poiPercBehind1 vs HeurBase3: [50, 50, 0]
```

```
poiPercBehind1 vs HeurBase2: [50, 50, 0]
poiPercBehind1 vs Aggressive: [50, 50, 0]
poiPercBehind1 vs Defensive: [0, 85, 15]
poiPercBehind1 vs bRANDy: [91, 9, 0]
Total score for poiPercBehind1 [291, 294, 15]

poiPercBehind2 vs HeurBaseK: [50, 50, 0]
poiPercBehind2 vs HeurBase3: [50, 50, 0]
poiPercBehind2 vs HeurBase2: [50, 50, 0]
poiPercBehind2 vs Aggressive: [50, 50, 0]
poiPercBehind2 vs Defensive: [0, 85, 15]
poiPercBehind2 vs bRANDy: [82, 18, 0]
Total score for poiPercBehind2 [282, 303, 15]

poiPercBehind5 vs HeurBaseK: [50, 50, 0]
poiPercBehind5 vs HeurBase3: [50, 50, 0]
poiPercBehind5 vs HeurBase2: [50, 50, 0]
poiPercBehind5 vs Aggressive: [50, 50, 0]
poiPercBehind5 vs Defensive: [0, 82, 18]
poiPercBehind5 vs bRANDy: [92, 8, 0]
Total score for poiPercBehind5 [292, 290, 18]

poiPercAhead1 vs HeurBaseK: [50, 50, 0]
poiPercAhead1 vs HeurBase3: [50, 50, 0]
poiPercAhead1 vs HeurBase2: [50, 50, 0]
poiPercAhead1 vs Aggressive: [50, 50, 0]
poiPercAhead1 vs Defensive: [0, 85, 15]
poiPercAhead1 vs bRANDy: [87, 13, 0]
Total score for poiPercAhead1 [287, 298, 15]

poiPercAhead2 vs HeurBaseK: [50, 50, 0]
poiPercAhead2 vs HeurBase3: [50, 50, 0]
poiPercAhead2 vs HeurBase2: [50, 50, 0]
poiPercAhead2 vs Aggressive: [50, 50, 0]
poiPercAhead2 vs Defensive: [0, 82, 18]
poiPercAhead2 vs bRANDy: [91, 8, 1]
Total score for poiPercAhead2 [291, 290, 19]

poiPercAhead5 vs HeurBaseK: [50, 50, 0]
poiPercAhead5 vs HeurBase3: [50, 50, 0]
```

```
poiPercAhead5 vs HeurBase2: [50, 50, 0]
poiPercAhead5 vs Aggressive: [50, 50, 0]
poiPercAhead5 vs Defensive: [0, 86, 14]
poiPercAhead5 vs bRANDy: [91, 9, 0]
Total score for poiPercAhead5 [291, 295, 14]

poiCompulsiveAct1 vs HeurBaseK: [50, 50, 0]
poiCompulsiveAct1 vs HeurBase3: [50, 50, 0]
poiCompulsiveAct1 vs HeurBase2: [50, 50, 0]
poiCompulsiveAct1 vs Aggressive: [50, 50, 0]
poiCompulsiveAct1 vs Defensive: [0, 86, 14]
poiCompulsiveAct1 vs bRANDy: [89, 11, 0]
Total score for poiCompulsiveAct1 [289, 297, 14]

poiCompulsiveAct2 vs HeurBaseK: [3, 97, 0]
poiCompulsiveAct2 vs HeurBase3: [2, 98, 0]
poiCompulsiveAct2 vs HeurBase2: [20, 80, 0]
poiCompulsiveAct2 vs Aggressive: [0, 100, 0]
poiCompulsiveAct2 vs Defensive: [0, 86, 14]
poiCompulsiveAct2 vs bRANDy: [81, 19, 0]
Total score for poiCompulsiveAct2 [106, 480, 14]

poiCompulsiveAct5 vs HeurBaseK: [0, 100, 0]
poiCompulsiveAct5 vs HeurBase3: [0, 100, 0]
poiCompulsiveAct5 vs HeurBase2: [0, 100, 0]
poiCompulsiveAct5 vs Aggressive: [0, 100, 0]
poiCompulsiveAct5 vs Defensive: [0, 99, 1]
poiCompulsiveAct5 vs bRANDy: [70, 28, 2]
Total score for poiCompulsiveAct5 [70, 527, 3]

poiTabooAct1 vs HeurBaseK: [0, 100, 0]
poiTabooAct1 vs HeurBase3: [0, 100, 0]
poiTabooAct1 vs HeurBase2: [0, 100, 0]
poiTabooAct1 vs Aggressive: [0, 100, 0]
poiTabooAct1 vs Defensive: [0, 79, 21]
poiTabooAct1 vs bRANDy: [55, 45, 0]
Total score for poiTabooAct1 [55, 524, 21]

poiTabooAct2 vs HeurBaseK: [0, 100, 0]
poiTabooAct2 vs HeurBase3: [0, 100, 0]
```

```
poiTabooAct2 vs HeurBase2: [0, 100, 0]
poiTabooAct2 vs Aggressive: [0, 100, 0]
poiTabooAct2 vs Defensive: [0, 83, 17]
poiTabooAct2 vs bRANDy: [58, 42, 0]
Total score for poiTabooAct2 [58, 525, 17]

poiTabooAct5 vs HeurBaseK: [0, 100, 0]
poiTabooAct5 vs HeurBase3: [0, 100, 0]
poiTabooAct5 vs HeurBase2: [0, 100, 0]
poiTabooAct5 vs Aggressive: [0, 100, 0]
poiTabooAct5 vs Defensive: [0, 87, 13]
poiTabooAct5 vs bRANDy: [44, 55, 1]
Total score for poiTabooAct5 [44, 542, 14]
```

Appendix N: Data Appendix - CNNs mutated with layer-targeted noise

N.1 Mutated CNNs, vs each other

	vsMainPolicy1k	mutConv1SD100	mutConv1SD10	mutConv1SD1	mutConv1SDp1	mutConv2SD100	mutConv2SD10	mutConv2SD1	mutConv2SDp1	mutFc1SD100	mutFc1SD1	mutFc1SDp1	mutFc1SDp01	TOTAL (W, L , D)
vsMainPolicy1k	- - - - -	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[0, 100, 0]	[600, 600, 0]
mutConv1SD100	[50, 50, 0]	- - - - -	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[50, 50, 0]	[0, 50, 50]	[0, 100, 0]	[450, 700, 50]
mutConv1SD10	[100, 0, 0]	[100, 0, 0]	- - - - -	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[1050, 150, 0]
mutConv1SD1	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[100, 0, 0]	[100, 0, 0]	[0, 100, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[850, 350, 0]
mutConv1SDp1	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[100, 0, 0]	[0, 100, 0]	[0, 100, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[0, 100, 0]	[50, 50, 0]	[500, 700, 0]
mutConv2SD100	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[100, 0, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[150, 1050, 0]
mutConv2SD10	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[900, 300, 0]
mutConv2SD1	[50, 50, 0]	[50, 50, 0]	[0, 100, 0]	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[0, 100, 0]	[50, 50, 0]	[600, 600, 0]
mutConv2SDp1	[50, 50, 0]	[100, 0, 0]	[0, 100, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	- - - - -	[100, 0, 0]	[100, 0, 0]	[0, 50, 50]	[50, 50, 0]	[700, 450, 50]
mutFc1SD100	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	[0, 100, 0]	- - - - -	[0, 100, 0]	[0, 100, 0]	[0, 0, 100]	[0, 1100, 100]
mutFc1SD1	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	[0, 100, 0]	[50, 50, 0]	[0, 100, 0]	[100, 0, 0]	- - - - -	[50, 50, 0]	[50, 50, 0]	[450, 750, 0]
mutFc1SDp1	[50, 50, 0]	[50, 0, 50]	[0, 100, 0]	[0, 100, 0]	[100, 0, 0]	[100, 0, 0]	[0, 100, 0]	[100, 0, 0]	[50, 0, 50]	[100, 0, 0]	[50, 50, 0]	- - - - -	[50, 50, 0]	[650, 450, 100]
mutFc1SDp01	[100, 0, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[100, 0, 0]	[50, 50, 0]	[50, 50, 0]	[50, 50, 0]	[0, 0, 100]	[50, 50, 0]	[50, 50, 0]	- - - - -	[700, 400, 100]

N.2 Mutated CNNs, vs Heuristic Functions

```
vsMainPolicy1k vs HeurBaseK: [50, 50, 0]
vsMainPolicy1k vs HeurBase3: [50, 50, 0]
vsMainPolicy1k vs HeurBase2: [50, 50, 0]
vsMainPolicy1k vs Aggressive: [50, 50, 0]
vsMainPolicy1k vs Defensive: [0, 84, 16]
vsMainPolicy1k vs bRANDy: [90, 10, 0]
Total score for vsMainPolicy1k [290, 294, 16]

mutConv1SD100 vs HeurBaseK: [21, 79, 0]
mutConv1SD100 vs HeurBase3: [13, 87, 0]
```

```
mutConv1SD100 vs HeurBase2: [12, 88, 0]
mutConv1SD100 vs Aggressive: [14, 86, 0]
mutConv1SD100 vs Defensive: [0, 91, 9]
mutConv1SD100 vs bRANDy: [84, 16, 0]
Total score for mutConv1SD100 [144, 447, 9]

mutConv1SD10 vs HeurBaseK: [10, 90, 0]
mutConv1SD10 vs HeurBase3: [14, 86, 0]
mutConv1SD10 vs HeurBase2: [10, 90, 0]
mutConv1SD10 vs Aggressive: [9, 91, 0]
mutConv1SD10 vs Defensive: [0, 90, 10]
mutConv1SD10 vs bRANDy: [91, 9, 0]
Total score for mutConv1SD10 [134, 456, 10]

mutConv1SD1 vs HeurBaseK: [50, 50, 0]
mutConv1SD1 vs HeurBase3: [50, 50, 0]
mutConv1SD1 vs HeurBase2: [50, 50, 0]
mutConv1SD1 vs Aggressive: [50, 50, 0]
mutConv1SD1 vs Defensive: [0, 77, 23]
mutConv1SD1 vs bRANDy: [88, 12, 0]
Total score for mutConv1SD1 [288, 289, 23]

mutConv1SDp1 vs HeurBaseK: [50, 50, 0]
mutConv1SDp1 vs HeurBase3: [50, 50, 0]
mutConv1SDp1 vs HeurBase2: [50, 50, 0]
mutConv1SDp1 vs Aggressive: [50, 50, 0]
mutConv1SDp1 vs Defensive: [0, 92, 8]
mutConv1SDp1 vs bRANDy: [91, 9, 0]
Total score for mutConv1SDp1 [291, 301, 8]

mutConv2SD100 vs HeurBaseK: [0, 100, 0]
mutConv2SD100 vs HeurBase3: [0, 100, 0]
mutConv2SD100 vs HeurBase2: [0, 100, 0]
mutConv2SD100 vs Aggressive: [0, 100, 0]
mutConv2SD100 vs Defensive: [0, 96, 4]
mutConv2SD100 vs bRANDy: [43, 56, 1]
Total score for mutConv2SD100 [43, 552, 5]

mutConv2SD10 vs HeurBaseK: [0, 100, 0]
mutConv2SD10 vs HeurBase3: [0, 100, 0]
```

```
mutConv2SD10 vs HeurBase2: [0, 100, 0]
mutConv2SD10 vs Aggressive: [0, 100, 0]
mutConv2SD10 vs Defensive: [0, 82, 18]
mutConv2SD10 vs bRANDy: [68, 31, 1]
Total score for mutConv2SD10 [68, 513, 19]

mutConv2SD1 vs HeurBaseK: [38, 62, 0]
mutConv2SD1 vs HeurBase3: [39, 61, 0]
mutConv2SD1 vs HeurBase2: [35, 65, 0]
mutConv2SD1 vs Aggressive: [40, 60, 0]
mutConv2SD1 vs Defensive: [0, 85, 15]
mutConv2SD1 vs bRANDy: [80, 20, 0]
Total score for mutConv2SD1 [232, 353, 15]

mutConv2SDp1 vs HeurBaseK: [50, 50, 0]
mutConv2SDp1 vs HeurBase3: [50, 50, 0]
mutConv2SDp1 vs HeurBase2: [50, 50, 0]
mutConv2SDp1 vs Aggressive: [50, 50, 0]
mutConv2SDp1 vs Defensive: [0, 76, 24]
mutConv2SDp1 vs bRANDy: [90, 10, 0]
Total score for mutConv2SDp1 [290, 286, 24]

mutFc1SD100 vs HeurBaseK: [0, 100, 0]
mutFc1SD100 vs HeurBase3: [0, 100, 0]
mutFc1SD100 vs HeurBase2: [0, 100, 0]
mutFc1SD100 vs Aggressive: [0, 100, 0]
mutFc1SD100 vs Defensive: [0, 87, 13]
mutFc1SD100 vs bRANDy: [0, 89, 11]
Total score for mutFc1SD100 [0, 576, 24]

mutFc1SD1 vs HeurBaseK: [0, 100, 0]
mutFc1SD1 vs HeurBase3: [0, 100, 0]
mutFc1SD1 vs HeurBase2: [0, 100, 0]
mutFc1SD1 vs Aggressive: [0, 100, 0]
mutFc1SD1 vs Defensive: [0, 87, 13]
mutFc1SD1 vs bRANDy: [47, 53, 0]
Total score for mutFc1SD1 [47, 540, 13]

mutFc1SDp1 vs HeurBaseK: [10, 90, 0]
mutFc1SDp1 vs HeurBase3: [14, 86, 0]
```

```
mutFc1SDp1 vs HeurBase2: [16, 84, 0]
mutFc1SDp1 vs Aggressive: [17, 83, 0]
mutFc1SDp1 vs Defensive: [0, 86, 14]
mutFc1SDp1 vs bRANDy: [89, 11, 0]
Total score for mutFc1SDp1 [146, 440, 14]

mutFc1SDp01 vs HeurBaseK: [50, 50, 0]
mutFc1SDp01 vs HeurBase3: [50, 50, 0]
mutFc1SDp01 vs HeurBase2: [50, 50, 0]
mutFc1SDp01 vs Aggressive: [50, 50, 0]
mutFc1SDp01 vs Defensive: [0, 90, 10]
mutFc1SDp01 vs bRANDy: [88, 12, 0]
Total score for mutFc1SDp01 [288, 302, 10]
```

Appendix O: Data Appendix - Ground Truth

```
            5-MiniNoiseLayer3 7-LowNoiseLayer1 TOTAL (W, L , D)
5-MiniNoiseLayer3 - - - - -[550, 450, 0] [550, 450, 0] Pink
7-LowNoiseLayer1 [450, 550, 0] - - - - -[450, 550, 0] Pumpkin

5-MiniNoiseLayer3 vs HeurBaseK: [28, 972, 0]
5-MiniNoiseLayer3 vs HeurBase3: [17, 983, 0]
5-MiniNoiseLayer3 vs HeurBase2: [118, 882, 0]
5-MiniNoiseLayer3 vs Aggressive: [127, 873, 0]
5-MiniNoiseLayer3 vs Defensive: [9, 620, 371]
5-MiniNoiseLayer3 vs bRANDy: [978, 21, 1]
Total score for 5-MiniNoiseLayer3 [1277, 4351, 372](Pink)

7-LowNoiseLayer1 vs HeurBaseK: [41, 959, 0]
7-LowNoiseLayer1 vs HeurBase3: [28, 972, 0]
7-LowNoiseLayer1 vs HeurBase2: [131, 869, 0]
7-LowNoiseLayer1 vs Aggressive: [182, 818, 0]
7-LowNoiseLayer1 vs Defensive: [0, 725, 275]
7-LowNoiseLayer1 vs bRANDy: [922, 78, 0]
Total score for 7-LowNoiseLayer1 [1304, 4421, 275](Pumpkin)


            10-LowNoiseLayer3 4-MiniNoiseLayer4 6-MiniNoiseLayer2 8-LowNoiseLayer2 TOTAL (W, L , D)
10-LowNoiseLayer3 - - - - -[182, 818, 0] [217, 783, 0] [361, 639, 0] [760, 2240, 0] Lime
4-MiniNoiseLayer4 [818, 182, 0] - - - - -[499, 501, 0] [726, 272, 2] [2043, 955, 2] Red
6-MiniNoiseLayer2 [783, 217, 0] [501, 499, 0] - - - - -[641, 358, 1] [1925, 1074, 1] Blue
8-LowNoiseLayer2 [639, 361, 0] [272, 726, 2] [358, 641, 1] - - - - -[1269, 1728, 3] Lavender

10-LowNoiseLayer3 vs HeurBaseK: [3, 997, 0]
10-LowNoiseLayer3 vs HeurBase3: [7, 993, 0]
10-LowNoiseLayer3 vs HeurBase2: [79, 921, 0]
10-LowNoiseLayer3 vs Aggressive: [34, 966, 0]
10-LowNoiseLayer3 vs Defensive: [5, 557, 438]
10-LowNoiseLayer3 vs bRANDy: [916, 81, 3]
Total score for 10-LowNoiseLayer3 [1044, 4515, 441](Lime)
```

```
4-MiniNoiseLayer4 vs HeurBaseK: [45, 955, 0]
4-MiniNoiseLayer4 vs HeurBase3: [26, 974, 0]
4-MiniNoiseLayer4 vs HeurBase2: [151, 849, 0]
4-MiniNoiseLayer4 vs Aggressive: [163, 837, 0]
4-MiniNoiseLayer4 vs Defensive: [7, 647, 346]
4-MiniNoiseLayer4 vs bRANDy: [981, 19, 0]
Total score for 4-MiniNoiseLayer4 [1373, 4281, 346](Red)

6-MiniNoiseLayer2 vs HeurBaseK: [35, 965, 0]
6-MiniNoiseLayer2 vs HeurBase3: [32, 968, 0]
6-MiniNoiseLayer2 vs HeurBase2: [154, 846, 0]
6-MiniNoiseLayer2 vs Aggressive: [124, 876, 0]
6-MiniNoiseLayer2 vs Defensive: [11, 682, 307]
6-MiniNoiseLayer2 vs bRANDy: [981, 19, 0]
Total score for 6-MiniNoiseLayer2 [1337, 4356, 307](Blue)

8-LowNoiseLayer2 vs HeurBaseK: [7, 993, 0]
8-LowNoiseLayer2 vs HeurBase3: [4, 996, 0]
8-LowNoiseLayer2 vs HeurBase2: [144, 856, 0]
8-LowNoiseLayer2 vs Aggressive: [0, 1000, 0]
8-LowNoiseLayer2 vs Defensive: [0, 755, 245]
8-LowNoiseLayer2 vs bRANDy: [944, 55, 1]
Total score for 8-LowNoiseLayer2 [1099, 4655, 246](Lavender)
```