“Why did my AI agent lose?”: Visual Analytics for Scaling Up After-Action Review

Delyar Tabatabai, Anita Ruangrotsakun, Jed Irvine, Jonathan Dodge, Zeyad Shureih, Kin-Ho Lam, Margaret Burnett, Alan Fern, and Minsuk Kahng*

Oregon State University

Abstract

How can we help domain-knowledgeable users who do not have expertise in AI analyze why an AI agent failed? Our research team previously developed a new structured process for such users to assess AI, called *After-Action Review for AI (AAR/AI)*, consisting of a series of steps a human takes to assess an AI agent and formalize their understanding. In this paper, we investigate how the AAR/AI process can scale up to support reinforcement learning (RL) agents that operate in complex environments. We augment the AAR/AI process to be performed at three levels—episode-level, decision-level, and explanation-level—and integrate it into our redesigned visual analytics interface. We illustrate our approach through a usage scenario of analyzing why a RL agent lost in a complex real-time strategy game built with the StarCraft 2 engine. We believe integrating structured processes like AAR/AI into visualization tools can help visualization play a more critical role in AI interpretability.

1 Introduction

Analyzing errors in artificial intelligence (AI) systems has received much attention in human-AI interaction and visual analytics [1, 12, 21, 27, 31], and such analysis for *reinforcement learning (RL) agents* that make sequential decisions brings additional challenges [10, 13, 19, 28, 29]. Consider AlphaGo [26] or a logistics-planning agent. If it performs poorly, it is important to know why. However, it is a challenging task because it involves many complex decisions and is often specific to the domain. Thus, these analyses can be better performed by application domain experts who know the domain well, but they often do not have expertise in the underlying AI algorithms.

Our team recently developed a new structured *process* for domain experts to assess AI, called *After-Action Review for AI (AAR/AI)* [9, 18]. We derived it from *After-Action Review (AAR)*, made by the U.S. Army in 1970s [20], which has been successfully used for assessing human decisions in the military. A key idea of AAR/AI is that it provides a structured process through a series of steps a human takes to assess an AI agent. For example, for a decision an AI has made (e.g., re-routing a delivery truck), a user is first asked to (1) answer the question “*what was supposed to happen at this decision?*”, then (2) identify “*what actually happened at the decision*”, and then (3) describe “*why it happened*”, possibly with visual explanations. These steps can be repeated for other decisions made by the AI. The previous AAR/AI empirical studies where domain experts took steps to explicitly answer such questions showed a number of benefits. AAR/AI helped the participants build mental models of the AI, led them to consider a diversity of perspectives, and helped them gain a high-level understanding of the AI [8, 9, 18].

However, the AAR/AI prototype used in our empirical studies was not designed to scale to complex, large-scale real-world environments. In particular, the study focused on having participants use an explanation interface to analyze pre-selected decisions that were known to have issues or bugs [16, 18]. In real use cases, an AI agent makes a large number of sequential decisions, and a user must also determine which decisions to analyze, in order to gain the best insight into the agent (e.g., what reasoning flaws did it exhibit).

In this paper, we present a visual analytics approach that integrates the AAR/AI process for domain experts to analyze RL agents that operate in complex environments. We augment the AAR/AI process to be performed at multiple levels of granularity, such that users first explore the entire *episode* (i.e., game), then dive into detailed analysis of individual *decisions* with *explanations*. We illustrate our approach through a usage scenario of analyzing why an agent lost in a complex real-time strategy game built with the StarCraft 2 engine.

This work opens up an interesting direction to the field of visual analytics. Although most visualization tools for AI interpretability consist of many views [12, 28] and their users are often asked to freely explore them, the free-form exploration may not be optimal for domain experts who do not know every detail of the underlying AI systems [14]. Our structured process addresses this by guiding them to relevant information. With a growing trend in interactive articles (e.g., explanatory articles on Distill [11, 22], New York Times articles relying more on scrolling than advanced interactions [2]), we believe integrating processes like AAR/AI or guidance [4, 5, 7] into visualization tools can help visualization play a more critical role in AI interpretability.

2 Background: Environment and AAR/AI

2.1 The Domain

Our team built a custom real-time strategy (RTS) game, “Tug-of-War”, using the StarCraft 2 game engine [17]. In this game, two AI players, Friendly AI and Enemy AI, play against each other in the top and bottom “lanes” over the course of a maximum of 40 Decision Points (DPs), which occur every 30 seconds (Fig. 1). At each DP, each player selects an action based on their resources, the battlefield conditions and how much they have spent and earned prior to the current DP. Actions include purchasing troop production buildings in one of the lanes and/or purchasing Pylons to increase income. The three types of troops–*Marines*, *Banelings*, and *Immortals*–have different costs, and have a rock-paper-scissors relationship (e.g., Marines are effective against Immortals; Immortals against Banelings). A player wins by destroying one of the opponent’s two Nexuses within 40 rounds. If none are destroyed, the player with the lowest health Nexus lost.

2.2 Reinforcement Learning Agents

The Friendly AI’s decision is determined by our *model-based reinforcement learning* agent [17]. At each Decision Point, the agent ranks the possible actions for the current state based on their action values, which estimate the probability of each action leading to an eventual win. To estimate the action values, the agent internally creates a look-ahead tree (visualized in Fig. 3-C), where the root node represents the current state of the game and each of its

*e-mail: {tabatase, ruangroc, jed.irvine, dodgej, shureihz, lamki, burnett, alan.fern, minsuk.kahng}@oregonstate.edu

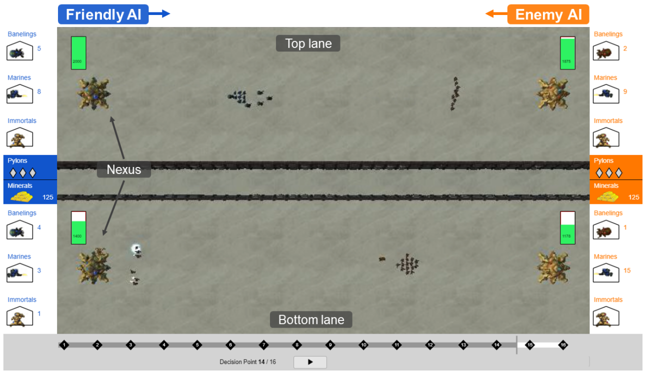
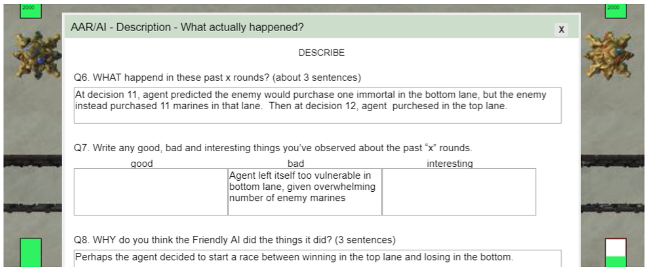


Figure 1: Our Tug-of-War game environment. Two AI players, Friendly AI (left) and Enemy AI (right), play against each other on top and bottom lanes by making a decision every 30 second over time.

children represents the predicted future state for each of the possible actions the agent can take. To decrease the complexity, only two look-ahead steps are computed. The agent constructs the tree, similar to MuZero [25], using predictions from the following three neural networks: (1) *Action-Ranking Function* which performs a fast prediction of the probability of an action in a state leading to a win and prunes low-scoring actions from the tree; (2) *Transition Function* which predicts the next game state given an initial state and actions proposed for the Friendly and Enemy players; and (3) *Leaf Evaluation Function* which returns a value estimate (probability of winning) for a game state. After building a tree, the leaf evaluations are propagated up the tree to compute the root action values.

2.3 AAR/AI Steps for Assessing the AI Agents

In our prior studies, we instantiated and evaluated an AAR/AI process for the domain and agent we described above. A participant watched a game replay and for every few Decision Points, they were provided with an *AAR/AI Prediction Questionnaire* to think about "*what is supposed to happen next*" by answering questions like "*Will the Friendly AI make any marines?*". Next, they continued watching the game replay to see *what actually happened* and whether their prediction was correct. They were asked to fill out a *Description Questionnaire* where they described what they observed and then reasoned about the AI's decision through questions like "*Why do you think the Friendly AI did the things it did?*" (see Fig. 2). Then, to help them further reason about the AI's behavior, they were asked to view and analyze an explanation presented in a graphical tree format showing what *actions* the AI considered and how the AI predicted future *states* if it took the actions. In our recent study [16], participants were presented with another questionnaire aimed at locating and describing *faults* they find in the AI through the explanation. For example, the AI might incorrectly predict a future that cannot happen, e.g., the health value of a Nexus increases over time (it must always monotonically decrease) because of an error in the AI's *Transition Function* component. The questionnaire consists of questions like "*Why did it happen*" and "*What changes would you make*". By explicitly writing down their answers to these questions, users can actively reason about the AI in a structured manner. Multiple qualitative and quantitative studies on AAR/AI indicate that it helps domain experts assess an AI by letting them build mental models, get a high-level understanding of the AI, and find more bugs [8, 9, 16, 18, 24].



AAR/AI - Description - What actually happened?

DESCRIBE

Q6. WHAT happend in these past x rounds? (about 3 sentences)

At decision 11, agent predicted the enemy would purchase one immortal in the bottom lane, but the enemy instead purchased 11 marines in that lane. Then at decision 12, agent purchesed in the top lane.

Q7. Write any good, bad and interesting things you've observed about the past "x" rounds.

good	bad	interesting
	Agent left itself too vulnerable in bottom lane, given overwhelming number of enemy marines	

Q8. WHY do you think the Friendly AI did the things it did? (3 sentences)

Perhaps the agent decided to start a race between winning in the top lane and losing in the bottom.

Figure 2: AAR/AI Description form helps users systematically reason about AI's decisions by answering a series of open-ended questions.

3 Scaling Up AAR/AI with Visualizations

This section describes scalability challenges we address in this paper and our approach to addressing them.

3.1 Design Challenges

The primary tasks we aim to help domain experts with are finding and understanding the reasons an AI agent failed (e.g. lost the game). In this paper, we address two scalability challenges that manifest in many real-world scenarios where reinforcement learning agents are used in sequential domains:

1. Long episodes may involve a large number of sequential decisions. How can we help users prioritize which decisions to analyze?
2. It would be overwhelming to show all the detailed information about an AI's reasoning process if many actions or multiple neural networks are involved. Showing an overview first, then allowing the user to drill down to details could be a better approach. What information should we show first and how much information should we show to non-AI experts?

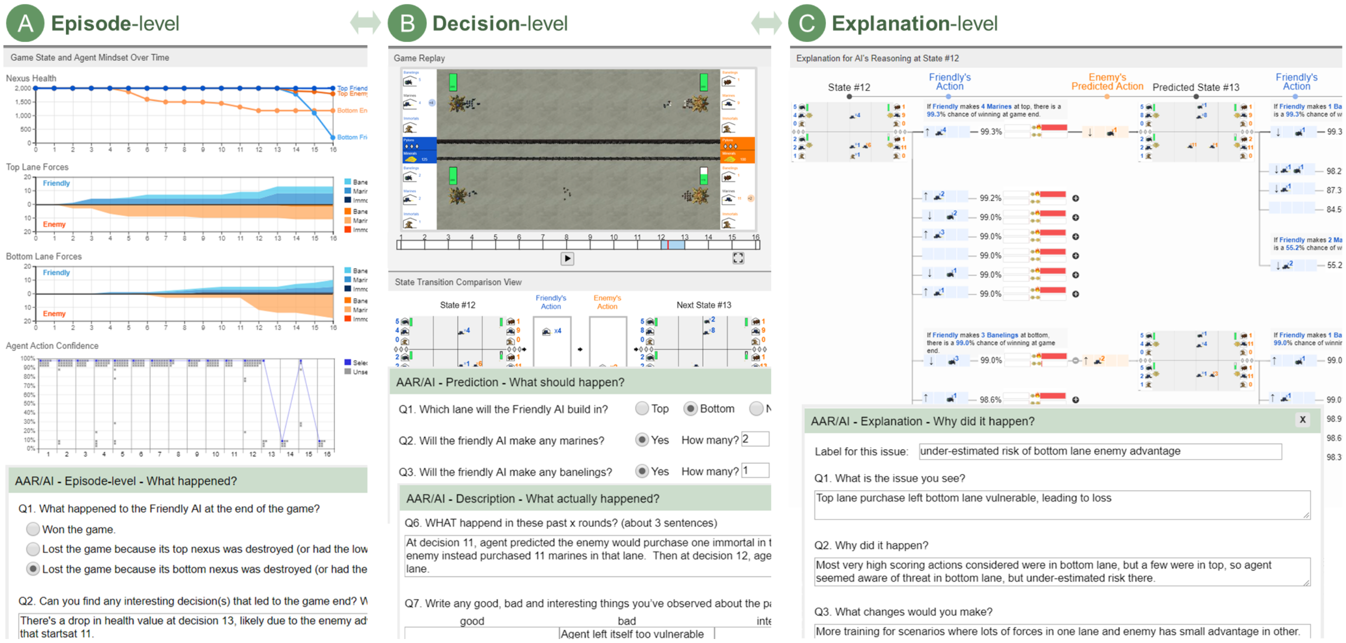
Our approaches to addressing each of the challenges are:

1. **AAR/AI at multiple levels.** We extend the AAR/AI process to be performed at multiple levels of granularity, so that users review the entire *episode* (i.e., game) first and then dive into further details of individual *decisions* and their *explanations*.
2. **Decision explanation with overview+detail.** Instead of visualizing the entire look-ahead tree and underlying neural networks, we visually summarize the list of *actions* and let users interact with each of them to see how the AI predicted future states for each action.

3.2 AAR/AI at Multiple Levels

Our approach augments the AAR/AI process by allowing the AAR/AI loop (e.g., "what was supposed to happen", "what happened", "why happened") to be performed not only at a decision-level but at different levels of granularity, specifically at the following three levels:

- **Episode-level:** A user is guided to get an overview of a game episode to learn how the game led to the loss (or win). They are asked to find interesting patterns over the timeline of the episode and identify decisions (i.e., "when") that are worth further investigation.
- **Decision-level:** For each decision identified, the user is guided to first *predict* "what decision was supposed to be made by an AI" based on the current state of the game, and then watch the game replay to see "what decision the user thinks the AI should have made" and *describe* it.
- **Explanation-level:** Finally, the user is provided with a visual explanation of the AI's decision to get a detailed picture of *why* the decision was made. They are asked to follow the AI's reasoning by exploring and navigating a look-ahead tree created by the RL agent. They can identify unexpected behaviors (e.g., unrealistic predictions) and reason about why.


A Episode-level
B Decision-level
C Explanation-level
Game State and Agent Mindset Over Time
Nexus Health
Top Lane Forces
Friendly
Enemy
Bottom Lane Forces
Agent Action Confidence
Game Replay
State Transition Comparison View
State #12
Friendly's Action
Enemy's Action
Next State #13
Explanation for AI's Reasoning at State #12
Enemy's Predicted Action
Predicted State #13
If Friendly makes 4 Marines at top, there is a 99.3% chance of winning at game end.
If Friendly makes 3 Banelings at bottom, there is a 99.0% chance of winning at game end.

AAR/AI - Episode-level - What happened?

Q1. What happened to the Friendly AI at the end of the game?

○ Won the game.

○ Lost the game because its top nexus was destroyed (or had the low

● Lost the game because its bottom nexus was destroyed (or had the

Q2. Can you find any interesting decision(s) that led to the game end? W

There's a drop in health value at decision 13, likely due to the enemy ad that startsat 11.

AAR/AI - Prediction - What should happen?

Q1. Which lane will the Friendly AI build in? ○ Top ● Bottom ○ N

Q2. Will the friendly AI make any marines? ● Yes How many? 2

Q3. Will the friendly AI make any banelings? ● Yes How many? 1

AAR/AI - Description - What actually happened?

Q6. WHAT happend in these past x rounds? (about 3 sentences)

At decision 11, agent predicted the enemy would purchase one immortal in t enemy instead purchased 11 marines in that lane. Then at decision 12, age lane.

Q7. Write any good, bad and interesting things you've observed about the pa

good	bad	inte
	Agent left itself too vulnerable	

AAR/AI - Explanation - Why did it happen? X

Label for this issue: under-estimated risk of bottom lane enemy advantage

Q1. What is the issue you see?

Top lane purchase left bottom lane vulnerable, leading to loss

Q2. Why did it happen?

Most very high scoring actions considered were in bottom lane, but a few were in top, so agent seemed aware of threat in bottom lane, but under-estimated risk there.

Q3. What changes would you make?

More training for scenarios where lots of forces in one lane and enemy has small advantage in other.

Figure 3: Domain experts can analyze complex AI agents with our visual analytics tool that integrates the AAR/AI process at three different levels. **A.** They can start with episode-level analysis to determine which decision to further explore, then **B.** take a detailed look into each decision by following the two-step AAR/AI process to formalize their understanding of AI, and then **C.** dive into AI's detailed reasoning process using the explanation that visualizes the AI's reasoning as an interactive look-ahead tree. The tool consists of multiple views, and while the users exploring each view, the AAR/AI forms serve as guidance for their analysis.

3.3 The New Visual Analytics Tool

The AAR/AI prototype we used in our previous studies contained only two views (game replay and explanation tree) because it was designed to focus on having participants use an explanation interface to analyze pre-selected decisions. We significantly redesigned it to support the new three-level AAR/AI process.

[Episode-level] Timeline view. Our new *timeline view* provides a starting point for a user to understand the overall game *episode* and prioritize which decisions to explore among many. The user can use charts that show changes of a few variables over time and fill out the AAR/AI "when" form. The choice of charts can be flexible depending on the domain, and we implemented four: a line chart showing Nexus health changes over time; two stacked area charts showing force composition over time for top and bottom lanes (e.g., number of Marines); and a sequence of dot plots (a unit visualization [23] of vertical histograms) showing the predicted win probabilities of the actions for the Friendly AI over time.

[Decision-level] Game replay and decision summary. Once the user determines which decision point to investigate, they can watch the replay of the game around the selected point with the video player with additional information about the state of the game (e.g., number of marines, Nexus's health value). While the user is watching the replay, they are asked to fill out the *Prediction* and *Description* forms mentioned earlier. In addition to the replay view, which was included in the previous version of the prototype, we have added a small view that summarizes each decision point for users to easily compare what was predicted by the AI and what actually happened so that they may skip watching the replay if they want to quickly scan the decision.

[Explanation-level] Explanation tree. The user can further investigate *why* an AI agent made a particular decision using the explanation tree. The previous version of our prototype showed an AI's decision as a look-ahead tree drawn as a basic node-link tree where each node corresponds to a game state (root as current state; child as a predicted next state if it takes an action) [18]. To support greater scalability and complexity, we redesigned the way this tree explains the AI's reasoning process. The new design supports overview+detail [6] by collapsing most actions (i.e., children) except the selected one shown at the top with a verbal summary of its eventual outcome (e.g., "*If Friendly AI purchases 4 Marines, there is a 99.3% chance of winning.*") and then allowing users to expand each action to see the detailed prediction of their future states (Fig. 3-C). In our new design, we also rotated the tree by 90 degrees from the initial version [18] so that the horizontal axis indicates time as a temporal flow (often represented horizontally), and each row represents an action (to accommodate many actions). We note that, as in the initial prototype, we do not show internal details of neural network models (e.g., how a predicted state is generated from a neural network model for the *Transition Function*), because non-AI expert users are unlikely interested in such details. Instead we focused on showing the AI's reasoning process at a higher-level via the look-ahead tree constructed from calls to the neural networks [9],

4 A Usage Scenario

This section presents a usage scenario based on what we found from our trained AI agent. Suppose a domain-knowledgeable user Jane (who is familiar with StarCraft 2) wants to analyze why a Friendly AI agent lost in a game by using our tool that integrates AAR/AI.

Determining Cause of Loss

Exploring the Timeline view for game overview. Jane starts searching for the cause of the agent's loss. A best case would be if she found where the agent makes a decision that defies her expectations and then leads to the loss. Based on what she was asked from the AAR/AI "when" form, she starts by looking at the *nexus health chart* (shown in Fig. 3-A) and notices a sudden drop in the Friendly AI's health for the bottom lane that caused it to lose, starting at Decision Point (DP) #13. Then, she focuses on force composition in the *top* and *bottom lane forces* charts. She sees the Friendly and Enemy AIs have fairly balanced forces over time in the top lane, however, the Enemy AI starts building up a force advantage in the bottom lane around at DP #11, which could logically lead to the Friendly agent's health decline that starts at DP #13. She describes this interesting observation on the AAR/AI form and also writes down her answer to the question "Why do you think it happened?" as "*The Friendly AI missed the threat in the bottom lane*", which helps her think about what to look for next.

Noticing surprising behavior. Jane wants more information on what happened at DP #11. From the *decision summary view*, she notices that the Enemy AI purchased 9 Marine units in the bottom lane. By clicking on the "Predicted" button she also sees that the agent predicted that the Enemy AI would purchase only a single Immortal unit, which is a big difference from the 9 Marines.

Predicting AI's decision. Jane wonders how the Friendly AI will respond to this surprise. By inspecting at the current state of the game, she notices that the Enemy AI's bottom lane nexus is the only one with low health, indicating the Friendly AI has made a lot of progress in damaging it. This, coupled with the addition of 9 enemy Marines in the bottom lane, gives the Friendly AI multiple reasons to counter in the bottom lane at the next DP. She documents this in the AAR/AI Prediction form (the first form in Fig. 3-B) that she expects the agent to purchase something in the bottom lane.

Describing AI's decision. Jane then plays the replay video to see whether the Friendly agent matches her prediction of a bottom lane action. She is surprised to see that the agent made a move in the top lane–a suspect strategic choice. The AAR/AI Description form guides Jane to describe this disconnect in a systematical manner. (Fig. 2). Jane is first asked to simply describe what she saw. The next question asks whether anything good, bad, or interesting happened, leading Jane to make a value judgement about what she saw: "*Agent left itself vulnerable in the bottom lane, given overwhelming number of enemy marines.*" Then, she is asked to speculate on why the agent did what it did: "*Perhaps the agent decided to start a race between winning in the top lane and losing in the bottom.*"

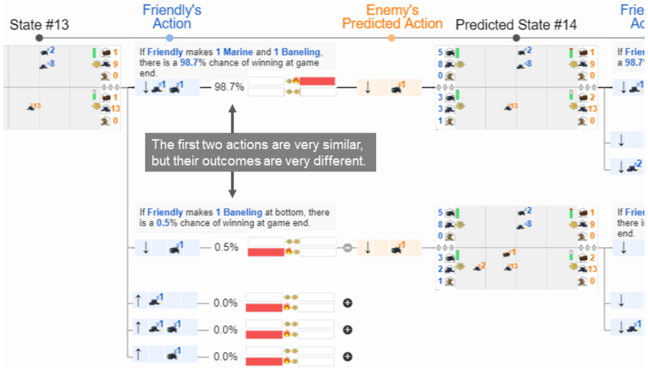
Analyzing the explanation. Now Jane wants to see if she can gain insights into the agent's thinking by using the *explanation tree* for DP #12 (Fig. 3-C). She first notices that most of the Friendly AI's action choices are scored very high – between 98 and 99%. She looks at the lane choice for these high scoring actions and notices that there are many bottom lane actions too. She thinks that the Friendly AI is aware of the threat in the bottom lane, but the presence of the few top lane choices suggests that it is underestimating the urgency of that threat. She opens the AAR/AI "why" form to record her thoughts about the agent's thinking as: "*Top lane purchase left bottom lane vulnerable, leading to loss.*". After she answers to the "Why did it happen?" question, she makes suggestions to AI engineers on how they might better train the agent and submits the form: "*More training for scenarios where it has lots of forces in one lane and enemy has small advantage in other.*"

Further Exploration for Spotting Bugs

Jane has successfully used the tool with AAR/AI to find a key strategy error made by the Friendly AI that might lead to the loss, but wonders if she can find other problems as well.

Deciding which decisions to explore. Going back to the charts, Jane recalls there was something interesting in the *action confidence chart* (i.e., distribution of the win probabilities for actions). Jane notices how through most of the game, the agent thinks that nearly all the moves it is thinking about will very likely lead to a win, but starting at DP #13 it starts to swing wildly back and forth. She decides to go straight to the *explanation tree view* to dig deeper.

Discovering anomalous patterns. Jane first notices that there are far fewer actions being considered by the agent (shown in Fig. 4). This is likely due to the agent not having much money to spend, so its purchase options are more limited. Jane sees the top ranked action has almost 100% confidence of a win if the agent makes that move, and the rest are pretty much 0%, which is very suspicious. Jane decides to expand the node associated with the second action of the explanation tree. One thing that popped out When she expanded this tree, she noticed the first two moves were nearly identical, but they yielded completely different outcomes. The only difference is that, in the top row, the agent is buying one additional marine unit. It so happens that in this game the marine units are the least powerful, so it seems odd that adding one marine would cause the chance of win to change so completely. Anything odd Jane sees in the explanation tree is worth reporting. On the AAR/AI "why" form. Jane describes the issue with the label: "*Very similar actions, very different outcomes*" Then she provides her answer to the "Why did it happen" question: "*Transition model seems over-sensitive to marine purchase in this scenario.*"



If Friendly makes 1 Marine and 1 Baneling, there is a 98.7% chance of winning at game end.

The first two actions are very similar, but their outcomes are very different.

If Friendly makes 1 Baneling at bottom, there is a 0.5% chance of winning at game end.

Figure 4: Explanation Tree for Decision Point #13 reveals a suspicious pattern in the AI's reasoning: the first two actions were nearly identical (i.e., only 1 Marine difference), but they yielded completely different outcomes (i.e., 98.7% vs. 0.5% chance of win).

5 Discussion and Future Work

We believe processes like AAR/AI have a strong potential in helping non-expert users understand and analyze AI using visualization. As visualization tools for AI become more complex to explain underlying AI systems, it becomes more challenging for users who are less familiar with AI to learn the full suite of features within a tool. AAR/AI can effectively address this challenge by serving as *guidance* (see Ceneda's review on guidance in visualization [5]). In addition, AAR/AI supports users' mental model building process which can be thought of as *learning* of the AI's mechanism [8]. This idea can potentially be applied to the recent literature on visualization for AI education designed for non-experts [12, 15, 22, 30]. Researchers have observed that standalone tools may not be the optimal medium for their learning, but interactive articles (e.g., those on Distill [22]) may foster better learning outcomes [11, 14]. Future work can study and evaluate different forms of guidance, processes, and their combination with free-form exploration, possibly by borrowing concepts from the education field, like our prior work on AAR/AI [18] measured people's level of understanding using Bloom's taxonomy [3], a well-known framework for categorizing different levels of *learning*. Future work can also include efforts on generalizing findings from tools designed for specific reinforcement learning environments to multiple different domains.

Acknowledgments

This work was supported by DARPA #N66001-17-2-4030. Any opinions, findings, conclusions, or recommendations expressed are the authors' and do not necessarily reflect the views of the sponsor.

References

[1] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019. doi: 10.1145/3290605.3300233

[2] D. Baur. The death of interactive infographics?, 2017. Dominikus Baur, Medium, `https://medium.com/@dominikus/the-end-of-interactive-visualizations-52c585dcafcb`, Accessed: June 17, 2021.

[3] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. *Taxonomy of Educational Objectives*. Longmans, Green and Co LTD, 1956.

[4] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017. doi: 10.1109/TVCG.2016.2598468

[5] D. Ceneda, T. Gschwandtner, and S. Miksch. A review of guidance approaches in visual data analysis: A multifocal perspective. In *Computer Graphics Forum*, vol. 38, pp. 861–879. Wiley Online Library, 2019. doi: 10.1111/cgf.13730

[6] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):1–31, 2009. doi: 10.1145/1456650.1456652

[7] C. Collins, N. Andrienko, T. Schreck, J. Yang, J. Choo, U. Engelke, A. Jena, and T. Dwyer. Guidance in the human–machine analytics process. *Visual Informatics*, 2(3):166–180, 2018. doi: 10.1016/j.visinf.2018.09.003

[8] J. Dodge, A. Anderson, R. Khanna, J. Irvine, R. Dikkala, K.-H. Lam, D. Tabatabai, A. Ruangrotsakun, Z. Shureih, M. Kahng, A. Fern, and M. Burnett. From "no clear winner" to an effective XAI process: An empirical journey. *Applied AI Letters*. (Early Access). doi: 10.1002/ail2.36

[9] J. Dodge, R. Khanna, J. Irvine, K.-H. Lam, T. Mai, Z. Lin, N. Kiddle, E. Newman, A. Anderson, S. Raja, C. Matthews, C. Perdriau, M. Burnett, and A. Fern. After-action review for AI (AAR/AI). *ACM Transactions on Interactive Intelligent Systems*, 2021. (To Appear).

[10] W. He, T.-Y. Lee, J. van Baar, K. Wittenburg, and H.-W. Shen. DynamicsExplorer: Visual analytics for robot control tasks involving dynamics and LSTM-based control policies. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 36–45. IEEE, 2020. doi: 10.1109/PacificVis48177.2020.7127

[11] F. Hohman, M. Conlen, J. Heer, and D. H. Chau. Communicating with interactive articles. *Distill*, 5(9):e28, 2020. doi: 10.23915/distill.00028

[12] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2674–2693, 2019. doi: 10.1109/TVCG.2018.2843369

[13] T. Jaunet, R. Vuillemot, and C. Wolf. DRLViz: Understanding decisions and memory in deep reinforcement learning. In *Computer Graphics Forum*, vol. 39, pp. 49–61. Wiley Online Library, 2020. doi: 10.1111/cgf.13962

[14] M. Kahng and D. H. Chau. How does visualization help people learn deep learning? Evaluating GAN Lab with observational study and log analysis. In *2020 IEEE Visualization Conference (VIS)*, pp. 266–270, 2020. doi: 10.1109/VIS47514.2020.00060

[15] M. Kahng, N. Thorat, D. H. Chau, F. B. Viégas, and M. Wattenberg. GAN Lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):310–320, 2019. doi: 10.1109/TVCG.2018.2864500

[16] R. Khanna, J. Dodge, A. Anderson, R. Dikkala, J. Irvine, Z. Shureih, K.-H. Lam, C. R. Matthews, M. Kahng, A. Fern, and M. Burnett. Finding AI's faults with AAR/AI: An empirical study. *ACM Transactions on Interactive Intelligent Systems*, 2021. (To Appear).

[17] K.-H. Lam, Z. Lin, J. Irvine, J. Dodge, Z. T. Shureih, R. Khanna, M. Kahng, and A. Fern. Identifying reasoning flaws in planning-based rl using tree explanations. In *IJCAI-PRICAI 2020 Workshop on Explainable Artificial Intelligence (XAI)*, 2021.

[18] T. Mai, R. Khanna, J. Dodge, J. Irvine, K.-H. Lam, Z. Lin, N. Kiddle, E. Newman, S. Raja, C. Matthews, C. Perdriau, M. Burnett, and A. 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)*, p. 465–476, 2020. doi: 10.1145/3377325.3377525

[19] A. Mishra, U. Soni, J. Huang, and C. Bryan. Why? why not? when? visual explanations of agent behavior in reinforcement learning. *arXiv preprint arXiv:2104.02818*, 2021.

[20] J. E. Morrison and L. L. Meliza. Foundations of the after action review process. Technical report, Institute for Defense Analyses, 1999.

[21] B. Nushi, E. Kamar, and E. Horvitz. Towards accountable AI: Hybrid human-machine analyses for characterizing system failure. In *Proceedings of the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2018.

[22] C. Olah and S. Carter. Research debt. *Distill*, 2(3):e5, 2017. doi: 10.23915/distill.00005

[23] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist. Atom: A grammar for unit visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3032–3043, 2018. doi: 10.1109/TVCG.2017.2785807

[24] S. Penney, J. Dodge, A. Anderson, C. Hilderbrand, L. Simpson, and M. Burnett. The shoutcasters, the game enthusiasts, and the ai: Foraging for explanations of real-time strategy players. *ACM Transaction on Interactive Intelligent Systems*, 11(1), 2021. doi: 10.1145/3396047

[25] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961

[27] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):353–363, 2018. doi: 10.1109/TVCG.2018.2865044

[28] J. Wang, L. Gou, H.-W. Shen, and H. Yang. DQNViz: A visual analytics approach to understand deep q-networks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):288–298, 2019. doi: 10.1109/TVCG.2018.2864504

[29] J. Wang, W. Zhang, H. Yang, C.-C. M. Yeh, and L. Wang. Visual analytics for rnn-based deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 2021. (Early Access). doi: TVCG.2021.3076749

[30] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, and D. H. Chau. CNN Explainer: Learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 2021. doi: 10.1109/TVCG.2020.3030418

[31] T. Wu, M. T. Ribeiro, J. Heer, and D. S. Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 747–763, 2019. doi: 10.18653/v1/P19-1073