Exploring Expertise through Visualizing Agent Policies and Human Strategies in Open-Ended Games

Steven Moore
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
StevenJamesMoore@gmail.com

John Stamper
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
jstamper@cs.cmu.edu

ABSTRACT

In this research, we explore the problem solving strategies of both humans and AI agents in the open-ended domain of video games. We utilize data collected from several human-level performing AI agents, that follow a given policy, and data from expert human players, that follow a set of strategies, for two Atari 2600 console games. We compare both types of data streams using a visualization technique to gain insights about how each player type, AI or expert human, go about solving the given games. Analyzing the action sequences of the two, we demonstrate how closely the agent policies resemble the real-world problem solving of a human player, and explore how we might extract human-level strategies for agent policies. We reflect on the benefits of using data from both AI agents and expert humans to instruct learners, model their behaviour, and how strategies may be more apparent and easier to adopt from human play. Finally, we hypothesize the benefits of combining both types of data for learning these complex tasks within open-ended domains.

Keywords

expertise, strategy, gameplay agent, visualization, t-SNE, deep reinforcement learning

1. INTRODUCTION

The process of building expertise, especially in complex tasks, has been an area of study for some time in education [11]. Issues related to the difficulty of data collection and storage, have been an impediment in the educational data mining (EDM) research community to explore many truly open ended complex tasks. In this research, we are taking steps to better understand how to collect, analyze, and gain a better understanding of complex environments where human expertise in the form of strategies may be used. We have selected a classic video game system environment based on the Atari 2600 console called the Arcade Learning Environment (ALE) [2]. ALE has generated a large amount of interest in recent years in the broader artificial intelligence and machine learning environments as a test bed for game playing agents. While the majority of work with this environment is focused on building general game playing agents, we have found the environment provides a useful test bed for understanding how humans learn and apply strategies, which can also be compared to agents. To this end, we are currently trying to understand how agents, that have met or exceeded human level capabilities at these games, encode strategies in their game policies, and how their strategies compare to expert human players.

In the development of these agents, it is the human encoding the strategy into the AI using their knowledge of the game. The majority of game-playing agents, however, make use of deep neural nets to develop their policies, which makes them black box and often difficult to interpret by a human. Recent work has looked at making policies developed this way programmatically interpretable, but much work remains for humans to be able to clearly articulate what many of these agents have learned from their training [31]. It is debatable if these deep reinforcement learning agents make use of explicit strategies as they execute their given policies. A recent approach uses saliency maps to highlight key decision regions for agents in ALE, and found that their agent for the Space Invaders videogame learned a sophisticated aiming strategy [12]. Another way to make policies less black box is to break the policy down into smaller subtasks that are comprised of a few actions that feed back into the overall policy [20]. These techniques of breaking down policies into smaller interpretable strategies and visually representing the mechanisms of an agent's policy are steps toward having humans learn strategies from agents, without directly encoding any into the agent itself.

While previous work continues to reduce the amount of training data required to develop successful agents via self-learning, others look to use human games to seed agents. One such study found that training on human data, they could achieve comparable scores to state-of-the-art reinforcement learning techniques and even beat the scores using just the top 50% of their collected data for more complicated games, such as pinball [16]. Combining a method that not only trains agents on expert human data, but also encodes their strategies into the form of an evaluation function, has the potential to yield successful agents that require less computational time while performing at greater levels than comparable agents.

Data from stochastic and adversarial domains remains challenging to mine, interpret, and visualize in a way that improves the understandability of the data. Video game data collected in the ALE is representative of this challenging domain, while also being open ended. Datamining and visualization techniques applied to such data can readily be leveraged for more traditional educational domains, such as solving a stoichiometry problem or completing a task in a physics simulator. One technique to help visualize such game data, in a way that enables us to make comparisons, is the use of t-distributed stochastic neighbor embedding (t-SNE) [19]. This is a technique used to visualize high-dimensional datasets and has previously been used a few times to visualize and interpret game data [22, 28]. By applying t-SNE for dimensionality reduction and visualization to data, similar clusters detailing potential strategies and policy enactment may emerge. Visually representing the mechanisms of an agent's policy can provide a step towards having humans learning strategies from these agents, gaining their expertise.

In complex tasks humans generate strategies which can be applied in many different situations. Combinations of strategies that lead to optimal outcomes can lead to expertise in a domain, although there is still no consensus among researchers as to what makes a person an expert and how expertise is defined. In this research we explore the interactions of policies and strategies, then look at how both relate to expertise in the context of these two games. Our long term goal is to see how humans can help teach agents and agents can help teach humans in a continuous loop hence the idea of "teachable humans and teachable agents." Specifically in this work, the main contribution is a start to this goal with a novel comparison of agent policies, generated with two different state of the art techniques on several complex game domains, and strategies generated from human players. We do this through the use of visualizing both types, expert human and agent, of collected gameplay data using t-SNE diagrams of the state spaces as a means to compare the two. We believe this work can help lead to a better understanding of human strategies and expertise, while also contributing to data mining techniques which can further be used in the context of explainable AI for educational systems. The visualization and comparison techniques used can be extended to more traditionally educational games, to gain a sense of any strategies being enacted. Additionally, it is beneficial to see if the agents are solving the game in a natural way, using a human-like strategy, as many similar systems are often designed to playtest such games and act as tutors to the users.

2. RELATED WORK

Expertise has been the subject at the crossroads of Psychology and Computer Science for some time. One of the first compiled works came from Glaser et al., *The Nature of Expertise* [11] explored a wide variety of domains from human typing to sports to ill defined domains. A key insight from this work is that in the early development of AI systems, expertise was tightly related to the concept of encoding human strategies into machines, such as early work involving chess players and intelligent tutors [4]. As work continued, there seems to be a drift from the Psychology field into architectures of cognition defined by ACT-R [1] and Soar [17] as examples. Computer Science moved towards agents and policy creation focusing early on reinforcement learning [29] and now advanced techniques built on deep learning [18].

2.1 Human Expertise and Strategies

The question of what exactly defines someone as an expert is still an open question and has a lot to do with the particular domain that is being studied. In chess, Chase and Simon posited that it takes 10,000 hours of study to become an expert in chess [4]. That number has also been suggested as the rough number of hours to become an expert musician [9] and is a general theory of expertise [10], although largely due to Simon's chess work.

In the case of learning systems, we often define mastery using some form of knowledge tracing. These systems often set "mastery" as a probabilistic value that a learner knows a particular skill. The value of mastery varies on skills and domains, but often a value of 90% or 95% are assumed to have achieved mastery [7].

Beyond measurements of expertise is it also important to qualitatively understand the strategies associated with expertise. Understanding strategies that are used to solve problems has also been explored in many domains. Tasks to elicit knowledge from experts, such as cognitive task analysis (CTA) have been used by cognitive scientists to better understand the 3 strategies that experts use, but may not explicitly recognize [6]. In a mathematics study on word problems where students were using cognitive math tutors, researchers noted three different strategies that students used to solve problems [8]. These strategies included (1) working backwards from the answer or unwinding, (2) plugging in values in a hill climbing method, and (3) using equations. With the correct structure of the problems these strategies could be explicitly identified.

2.2 Agent Expertise and Policies

Artificial intelligence has been used now for decades to create agents that mimic human behavior. These agents are generally driven by a policy created by some form of machine learning such as reinforcement learning [29]. The policy tells the AI agent what to do given a certain set of conditions. This is most often defined as a state-action graph that suggests the best possible next action for an agent assigned to a given state [27].

In education, agents driven by policies have long been a foundational part of data-driven intelligent tutors and adaptive learning. Work has been done in modeling learning at a partially observable Markov decision process (POMDP), and using a policy generated to predict what a student knows and what the next best instructional lesson is for a particular student [24]. Other research has been done using reinforcement learning with a focus on what pedagogical action would be best to use for a student when multiple actions are available [5]. Most closely associated with the research we are doing is working on the automatic generation of hints and feedback [25, 26]. This work uses state graphs and reinforcement learning to identify the best path for solving problems and using the state features of the next best state to generate a just in time hint, like the next optimal move in a game [23]. This type of feedback can lead the student down a better path for learning.

2.3 Comparing Human & Agent Gameplay

Visualizations of gameplay data are widely popular, often being used by players to compare their performance against others and to make sense of how they played the game [32]. For instance, heat maps have been used by players to refine gameplay strategies, providing insights into popular areas about a game's environment [15]. In a similar vein, saliency maps have been applied to gameplay from agents, acting as heat maps for activations in their neural nets [12]. From such visualizations, it became clear that the agent was enacting a form of a strategy around aiming, as a human player would do. Another use of saliency maps, combined with t-SNEs, looked to describe the policies agents were using [34]. This was done to not only make the agents less black-box and understandable, but to see if they followed any set strategies.

Many games are making use of such game-playing agents and procedurally generated content methods to develop both the game environment and to play-test the games [13]. Much like the agents developed for the ALE, these game-playing agents play a game in order to find any bugs or areas of improvement. With such large amounts of data coming from even the most simple games, many tools have been developed to assist in the visualization and analysis process [33]. Using visualizations is one way to gain insights into any human-like strategies being enacted by such agents. This is important as an agent might not be of much use if it plays the game, but not in a way that a human user does. In open-ended games with a massive state space, mimicking as close to human play as possible helps to provide the most accurate data and bug testing from the agent.

3. METHOD

3.1 Environment & Games

The Arcade Learning Environment (ALE) provides a framework consisting of over fifty Atari 2600 games that can used to evaluate competency in deep reinforcement learning (DRL) agents and other types of AI [2]. Despite having a limited amount of input, a fire button and four directional controls, many of the games consist of complex tasks in open-ended worlds, making them a fitting testbed for DRL agents. Using the ALE, we focused on gameplay from two distinct games for the Atari 2600. The first game is Space Invaders, which is one of the simpler games for the system, consisting of just four non-combinational inputs. The second game is Seaquest, which incorporates all input combinations available for the Atari 2600, making it a much more complex and challenging game for both humans and agents.

3.1.1 Space Invaders

In the game Space Invaders, depicted in Figure 1, the player or agent controls a ship at the bottom of the screen that can navigate along a single dimension of left or right. The goal of the game is to destroy all the enemy units above the user's ship, gaining points for each enemy destroyed, while also avoiding any projectiles from them. If the player is struck by an enemy projectile they lose one of their three lives. To destroy these enemy units, the player's ship can fire a projectile that goes directly up, damaging or destroying an enemy unit on contact. Additionally, the player can hide behind three objects at the bottom of the screen to avoid the enemy fire. The only valid controls for this game are left and right to move the player agent and the fire button to shoot.

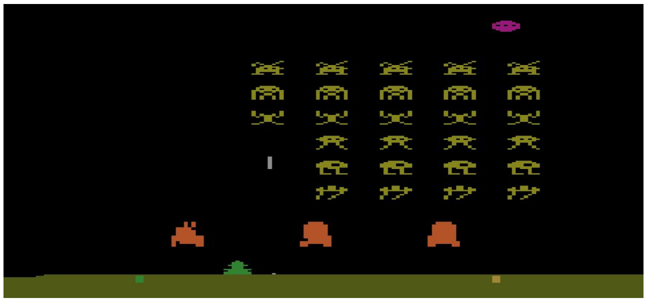

Figure 1. In Space Invaders the player controls the green ship at the bottom of the screen and must shoot the invaders that proceed left, then down, then right.

3.1.2 Seaquest

In the game Seaquest, depicted in Figure 2, the ultimate goal is to retrieve as many scuba divers from under the water as possible. The player or agent controls a submarine that can navigate in all directions around the screen and faces the front of the ship in the direction of movement, either right or left. This submarine has an oxygen tank gauge that slowly diminishes over time, the player must surface their ship at the top of the screen to refill it. As they navigate around the screen, collecting the divers, they also must dodge enemy ships and sharks that navigate across the map. If their submarine collides with an enemy unit or the oxygen gauge reaches zero, they lose one of their three lives. To combat these enemies, they are able to shoot a projectile from the front of the ship, which damages or destroys these enemy units. Killing an enemy results in a point increase, but the main increase in points comes from saving the divers. In order to receive points for the collected divers, the submarine must surface by navigating to the very top of the map. All valid button combinations for the Atari 2600 controller work for this game, such as up-left-fire, right-fire, and down.

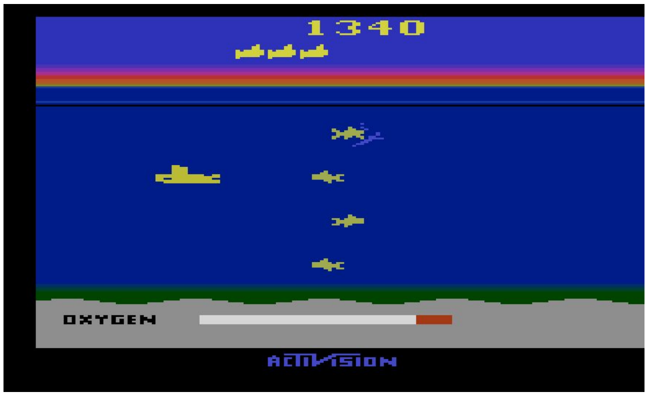

Figure 2. In Seaquest the player controls the yellow submarine, collects the scuba divers, and shoots or avoids the enemies.

3.2 Agent Dataset

As the ALE provides a framework for testing DRL agents, we selected two higher performing agents implemented in the environment using value-based DRL algorithms. The first agent utilizes a Deep Q-network (DQN) and has achieved a level comparable to a human professional in almost fifty games, including the two we investigate [21]. Our second is an agent known as Rainbow, which is built upon a DQN variant and has achieved even greater scores across the same Atari 2600 games [14]. We selected the DQN agent as it is often cited as a baseline for this domain. The Rainbow agent was selected for its high scoring performance, while still mimicking human play when observed. For instance, Rainbow will move the player avatar about the screen in Seaquest, rather than stay at the very bottom of the screen to avoid enemies, as some agents do.

The data for both of these agents come from the benchmarks used in the *Atari Zoo*, an open-source set of trained models for six major DRL algorithms at varying benchmarks, collected from the ALE [28]. Other DRL algorithm agents implemented in *Atari Zoo* perform at lower levels than DQN and Rainbow, while not mimicking human gameplay, such as A2C [22]. For this reason, we did not select those agents, as we wanted high performing ones for both games.

Table 1 shows the max score achieved per Space Invaders game for the Rainbow agent, DQN agent, and expert humans. The cells with multiple scores in them indicate the agent or human lost all lives during that session and restarted play within the limited amount of frames recorded. Thus, a single score indicates the agent or human did not lose all of their lives during the recorded play. Table 2 shows similar information, but for the game plays from Seaquest. In particular for the DQN agent, as shown in the second game play, the five scores low scores indicate the agent lost all lives and had to restart play five times in the allotted steps.

Table 1. The highest score(s) achieved for the two agents and expert human in the collected Space Invaders data over three different game plays.

Space Invaders Game	DQN	Rainbow	Human
1	2380	1805,990	1685
2	1495, 600	3750	1745
3	1345, 830	3845	1845

Table 2. The highest score(s) achieved for the two agents and expert human in the collected Seaquest data over three different game plays.

Seaquest Game	DQN	Rainbow	Human
1	800,1400	4960	12590
2	60, 60, 60, 60, 100	5020	14220
3	3900, 500	7840	16880

3.3 Expert Human Dataset

Using the ALE, we collected expert human data for both Space Invaders and Seaquest. To collect the human game play data, we modified the ALE code to record the RAM state at each frame of gameplay, so that it could be compared to the agent data from the *Atari Zoo*. Using Atari 2600 data collected from the Atari Grand Challenge project as a baseline for Space Invaders, our collected expert human data ranks in the top 1% based on scores [16]. We were unable to use the collected data from the Atari Grand Challenge, as we needed the RAM states in order to visualize the data in the *Atari Zoo*.

3.4 Visualizing

A popular technique used for dimensionality reduction and visualization of high-dimensional data used with large reinforcement learning datasets is t-SNE [22]. It provides a way to plot the data, from both agents and humans, along varying dimensions, clustering the related frames to one another. Our data, for both agents and humans, consisted of the Atari RAM representation, which is the same across agent algorithms and runs, but distinct between the games. Traditionally, the use of t-SNE embeddings are for a single high-level representation of an agent. However, since our datasets are all from the Atari RAM representation, this enables us to make comparisons between different runs of an agent for the same algorithm and runs from different DRL algorithms. As these datasets were quite large for both games, we pre-processed them using Principal Component AnalysiS (PCA) to a dimensionality of 50, then followed that with 300 t-SNE iterations with a perplexity of 30 [30]. Note that t-SNE positions the points on a place such that the pairwise distances between them minimizes a certain criterion. As a result, the axes can not be labeled with a specific unit, due to the high dimensional nature of the data.

Utilizing the code provided from the *Atari Zoo* [28], we are then able to visualize the processed agent and human data in a t-SNE embedding with associated screenshots. The points in the resulting t-SNE embeddings represent a separate frame from the agent or human. They are colored corresponding to their given source and the transparency is used to indicate score, with a darker color indicating a higher score. The clustering of the points help to indicate the distributions of states, corresponding to behaviour, the agent or human visited. Additionally, the points can be clicked on to view a screenshot of the game. This provides another metric for analyzing agent-collected data, in addition to providing a means of comparison to our collected human data.

4. RESULTS

4.1 Space Invaders

Plotting the DQN, Rainbow, and expert human data from Space Invaders via t-SNE, we can see both similarities and differences in the clustering. Figure 3 depicts a t-SNE embedding of nine Space Invaders games in total, three from each agent and the expert human. The agent data, green depicting Rainbow and blue for DQN, overlaps more throughout the graph than the human data points, represented by red. A majority of the human data clusterings are on the bottom half of the t-SNE, where there only appears to be a single Rainbow and DQN cluster. There is an equal separation of high scoring points, depicted by darker shades of the color, for all three parties. High scoring human points of dark red are scattered about, while the dark blue DQN data is grouped toward the upper center. Above that is the dark green Rainbow data, that is grouped between the 100 and 150 points of the y-axis. Ultimately while there is similar clustering of the Rainbow and DQN agents across all three games, it does not hold true for when the game is coming to an end and a higher score has been achieved. Additionally, regardless of the game's score, the human data does not seem to have much overlap with either agent.

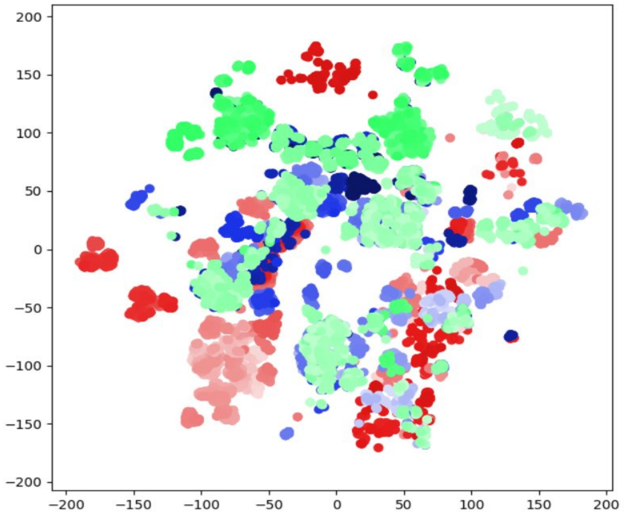

Figure 3. Two-dimensional t-SNE embedding of Space Invaders gameplay collected from three games using the Rainbow agent, depicted in green, three games from the DQN agent, depicted in blue, and three games from the expert human, depicted in red, for nine games in total.

To further identify any interesting clustering of the points, we selected a single game play from the two agents and the human data, so the t-SNE would show one from each for a total of three, instead of the aforementioned nine. The resulting t-SNE for this is shown in Figure 4, along with screenshots that are representative of the major clusters. We included screenshots for six clusters, two from each, that are darker in color corresponding to a higher score and being further along in the game. Since this depicts a later point in the game, any key moves or strategies are more visible since they have had time to be enacted. With a single game for each agent or human depicted, the representative clusters stand out even more.

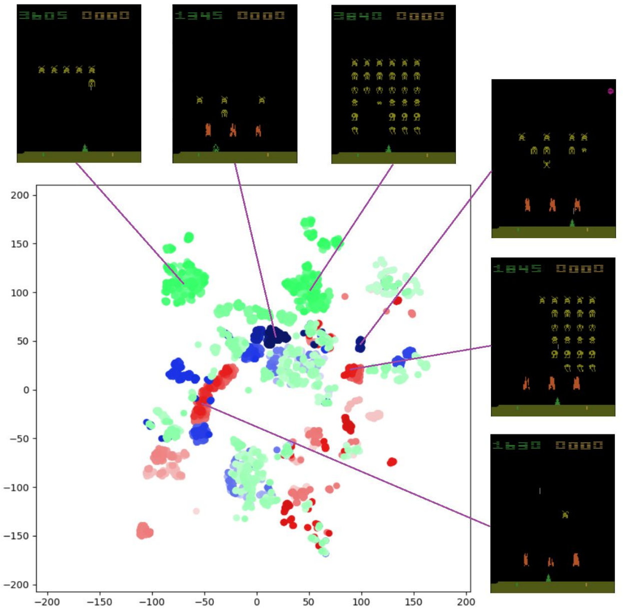

Figure 4. A t-SNE for a single game of Space Invaders from the green Rainbow agent, blue DQN agent, and red expert human with screenshots depicting the largest and high scoring clusters.

4.2 Seaquest

Following the same steps of the Space Invaders data, we plotted the collected Seaquest gameplay data via t-SNE. Figure 5 depicts the t-SNE embedding of nine Seaquest games in total, three from each agent and three collected from the expert human. Similar to the Space Invaders t-SNE embedding, the two Rainbow and DQN agents, represented by green and blue respectively, overlap more with one another than the expert human data, represented by the red points. However, all three types of points about the diagram are much less clustered into groups and more spread out throughout the given range, indicating a greater variance of game states between the two agents and the expert human. One notable clustering resulting from all nines games is a grouping in the center, where the Rainbow agent, shown in green, almost perfectly overlaps the DQN agent, shown in blue. For this cluster, the points for both agents are also darker, indicating they are for higher scoring states that occur later during the game play.

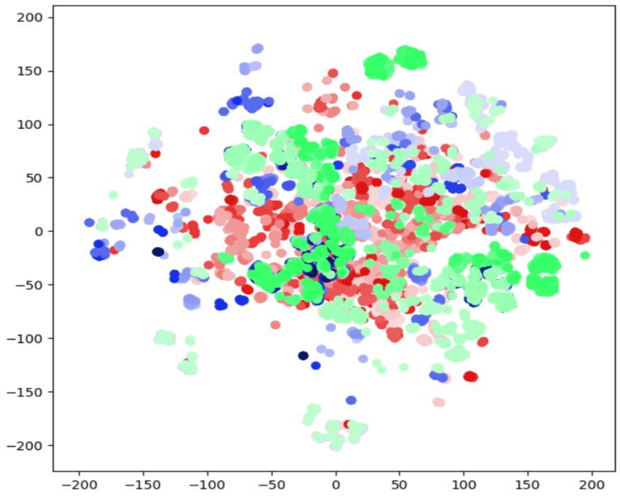

Figure 5. Two-dimensional t-SNE embedding of Seaquest gameplay collected from three games using the Rainbow agent, depicted in green, three games from the DQN agent, depicted in blue, and three games from the expert human, depicted in red, for nine games in total.

As the resulting data appears to be fairly scattered for all nine games of Seaquest, we selected just the third play through for both agents and the humans and displayed it via t-SNE, shown in Figure 6. With just a single game from each source displaying, several clusterings became more apparent. The Rainbow agent has three distinct clusters, two of which overlap with the DQN agent. Screenshots from these two clusters depict the player unit, the yellow submarine, towards the center of the screen with no enemies around. The representative screenshots depicting the expert human data, via the red points, show the shit less towards the center and with more enemy units about. This suggests a potential difference in gameplay between the agents and the human, that we elaborate on in the following discussion.

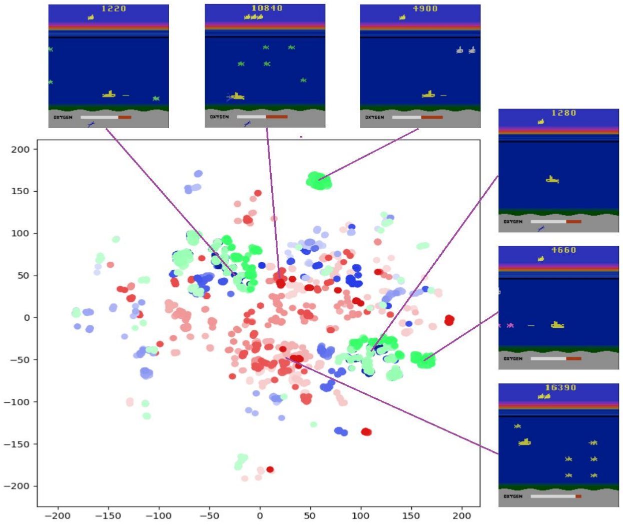

Figure 6. Representative screenshots for the various clusters about the t-SNE embedding for the third play of Seaquest from each agent and human dataset. Green points represent the Rainbow agent, blue points for the DQN agent, and red ones for human.

5. DISCUSSION

Plotting the game data via t-SNE provides a concise visualization of such high-dimensional data. However, they are only beneficial to us if their clusterings detail any patterns that might be indicative of a strategy or interesting behaviour. One immediate clustering that caught our eye was for the t-SNE depicting a single game of Space Invaders for the two agents and humans. As Figure 7 shows, there is a clear dark red cluster of expert human data toward the center, indicating there are many similar game states here and ones with a high game score comparatively. Examining the points on this curving cluster, we noticed the screenshots representing the game states at the time had a clear similarity. The states in this cluster were for when a single enemy ship was left on the map, the point right before the player can advance to the next stage. It became clear that human player had difficulty hitting the last few enemies, as they move fast and requires precise aiming when there are not many left. A nearby cluster from the Rainbow agent, represented in green and highlighted in Figure 7 too, depicts a similar set of states. However, there are not as many points for the agent in this set of states as there are the human, suggesting the agent can more accurately hit the fast moving last remaining enemies.

While this is not a particular strategy, it does provide insights into similar difficulties both agent and human have in the game. It also aligns with the maximum scores both agent and human achieved, as the agent spent less time on this phase and could advance through the game more rapidly, achieving a higher score in the allotted time, which one might equate to expertise in this domain. It is a case where the Rainbow agent is reflecting a difficulty also encountered by the human. If this was in the context of an educational game, we could use the agent's data to gain insights into where a hint or other feedback might be the most optimal, as it is a clear point of difficulty. Additionally, the DQN agent did not demonstrate such difficult. If just the DQN agent's data was used for such playtesting, this area of struggle may have been missed altogether. This insight was provided through a brief visual inspection enabled via t-SNE, that may not be as readily clear from parsing log data.

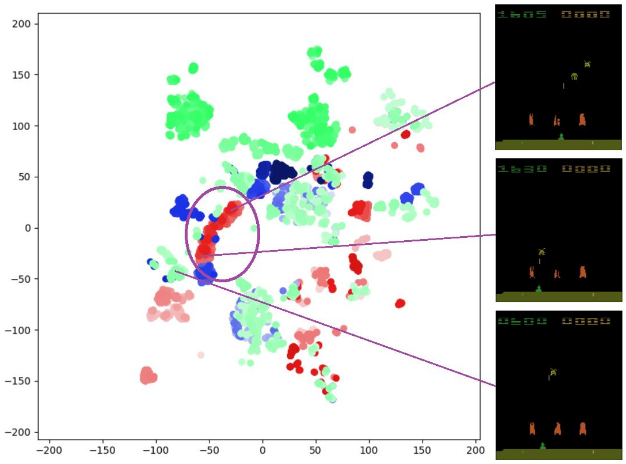

Figure 7. A long clustering of red points, representing human data, with screenshots from the top and bottom, showing a pattern of the player attempting to destroy the last remaining enemies. The nearby green points, for the Rainbow agent, represents a similar clustering as the agent finishes the final enemy.

Another analysis of the same t-SNE plot for a game of Space Invaders provided insights into a distinct shooting strategy the two agents and human each had. When we inspected clusters and points that represented the game at a halfway point, where half the enemy units on the screen were destroyed and the other half alive, we noticed an interesting pattern in the configuration of the remaining enemies. As Figure 8 shows, the Rainbow and DQN agents target enemies either horizontally across the bottom or in a diagonal pattern. However, the expert human destroys the enemy units starting from the left column and working right. While each of these represents a different shooting strategy for the given player, the expert human's strategy is debatably the most optimal. For Space Invaders, the enemy units move about the screen horizontally, and once they reach the edge of the screen they move down a single row, and continue moving the opposite horizontal direction. This means that if there are few enemy columns, it takes the enemies a greater time to horizontally traverse, allowing the player more time to fire at them.

There are trade-offs for this strategy though, as if the bottom row of the enemy units comes into contact with the ground, the game is over. This may be the reason why the deep reinforcement learning trained agents shoot in a horizontal or diagonal pattern, so that they keep the bottom row higher up to avoid the game over condition, something they must have encountered quite often during their early training phases. However, this does not translate to an optimal strategy as the expert human data reveals. Teaching a player the game using such agents could lead to the adoption of this firing strategy, which would be suboptimal compared to that of the human's. Such a case could also readily apply to more educational game contexts, as an agent or tutor that learned to play or solve the problem may be doing so in a non-optimal way compared to that of an expert human. Even though the "score" for a given game is greater, ultimately learning the better strategy would have a greater pay off in the long run.

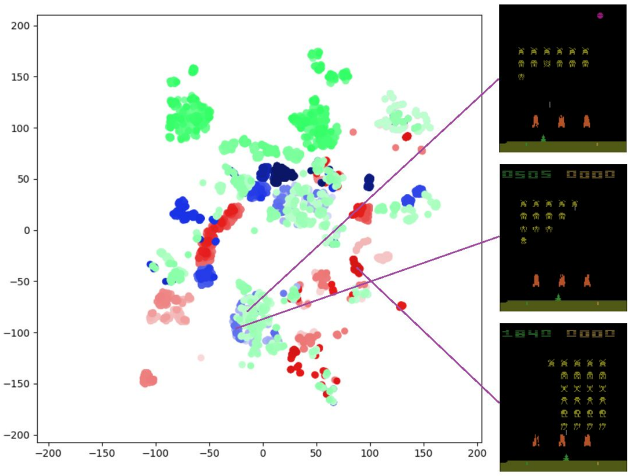

Figure 8. The DQN and Rainbow agent cluster towards the bottom shows that half way through the game, they keep the enemies clustered. This can be observed from the human data, however instead of going across the bottom or diagonal, they start from the left most column and work right.

Examining the t-SNE for a game of Seaquest also revealed insights into the differing navigational strategies used by both agent and human. For this t-SNE plotting, there was less clustering compared to the Space Invaders ones. However, as we investigated the different points and viewed the screenshots for representative states, a pattern with the player-controlled submarine emerged. For both the DQN and Rainbow agent, the submarine remained towards the bottom of the map and stayed centered on the y-axis, unless they were briefly moving to rescue a diver. However, the human points showed the submarine in a variety of positions that were far from the center or bottom axis, even without the presence of these scuba divers. As Figure 9 depicts, the human made use of more free moving navigational behavior, traversing the entire map and getting towards the edges to allow themselves more time to position and fire at enemies. The agents, who presumably had better accuracy from their mass training, could remain towards the center and only leave the bottom when they had to move upwards to fire at an enemy or surface for oxygen.

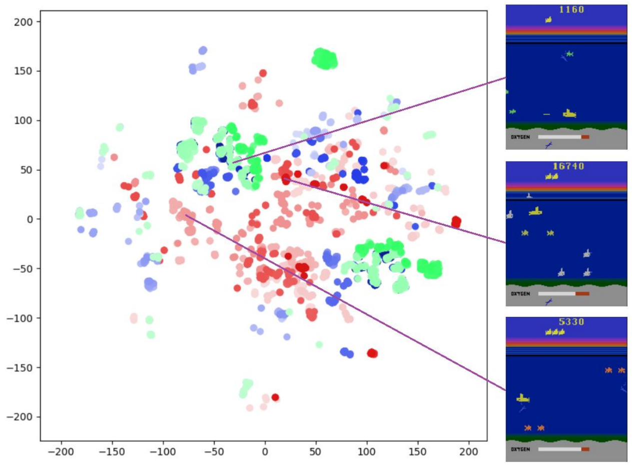

Figure 9. The red human data shows that they move about the screen more compared to the agents, depicted in blue and green, who most often end up in the center of the y-axis, particularly towards the bottom.

Similarly to the Space Invaders strategy of creating different enemy configurations, the agents for Seaquest had their own strategy that differed from the expert human gameplay. In this game’s context, there is not necessarily a clear benefit of one navigational strategy over the other. However, the one used by the agents might be better suited for a player who has better aim and does not need to get their avatar close to the enemy units. If a human user learned from the actions of these agents though, they might not move around the map as the expert human gameplay did. While not necessarily impacting the score, it could impact their enjoyment of the game, as they will be making less movements and have less control over their avatar, compared to treating it as basically fixed along the y-axis. This is another consideration of using just an agent to playtest or learn from, as even when it might not impact performance, other factors like enjoyment might be impacted from the enactment of certain agent performed strategies.

This research represents our initial exploratory work into understanding expertise of complex tasks in open ended domains using a combination of human and artificial intelligence agents. We have begun by plotting expert human and human-level agents using t-SNEs to provide a way for us to visualize data. We can see from the plotted t-SNEs that expert human data does have some overlap with data from high performing DRL agents, however, gaps exist where humans clusters are far away from the agent data. Nevertheless strategies from both human and agent data emerge in the visualizations and allow for some interesting comparisons between the two. There are clear implications of using just an agent’s gameplay, as the enacted strategies may be optimal, but limiting to a user’s play. They also might demonstrate a clear strategy, such as the firing configuration in Space Invaders, yet such a strategy could actually be sub-optimal for a human to enact.

While these two games are not traditional educational ones, the implications of the techniques used and insights gained are still applicable to ones in such a context. Eliciting strategies, regardless of coming from an AI system or human, is challenging and such visualizations provide one way to search for and understand them. At present, the use of agents using similar mechanisms and reinforcement learning methods to solve problems then instruct students agents [3] could benefit from the use of t-SNE visualization of the collected data. They want to ensure the strategies and suggested instruction are optimal, while remaining natural as a human would act. As it is not useful if a human cannot enact a particular suggested strategy, due to an agent having different control during the training process, such as access to frame-by-frame data in the game, causing it to have greater accuracy.

6. CONCLUSION & FUTURE WORK

Our primary goal in this work is to explore expertise, in this case in the context of games. In such games, prior work often uses the score as a measure of how expert a player, either human or agent, is at the game. We believe in addition to the score, the strategies used to solve the game impact how expertise, in this domain, can be quantified. To gain insights into such strategies we visualized gameplay data of a high scoring and long time playing human, deemed an expert, and high scoring agents gameplay data via t-SNE. Analysis of the resulting t-SNEs yielded insights into both shared and differing strategies the two parties had. Even between agents, there existed similar and dissimilar strategies, in addition to their score variance. Taking into account these gameplay differences and how realistic an enacted strategy might be for a human to learn from or mimick is important for game and tutor developers to keep in mind when using agents as playtesters or instructors. A strategy might be seem beneficial, yet compared to a different one it may not be as optimal nor practical for a player or learner to utilize in their own gameplay.

As we continue this work, we want to extend it to more games other than Space Invaders and Seaquest, particularly ones in the educational space that also have accompanying agent-collected data. Further inspection remains to be done to draw more strategies from the accompanying visualizations. Following this, we will further look into how they cluster, indicating the performance of similar strategies based on their policies. One key area we plan to explore is adding a temporal aspect to the t-SNE graphs. Although not represented in our current visualizations, we do have the screenshots numbered temporarily, so we expect that we can connect the paths to show the progression of game play. Additionally, visualizing novice human data, in addition to the expert and agent data, could provide useful strategy comparisons. This could help developers of educational games find where their novice learners seem to struggle the most, from a visual standpoint.

7. ACKNOWLEDGMENTS

The research reported here was supported in part by a training grant from the Institute of Education Sciences (R305B150008).

Opinions expressed do not represent the views of the U.S. Department of Education.

8. REFERENCES

[1] Anderson, J.R., Matessa, M. and Lebiere, C. 1997. ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*. 12, 4 (1997), 439–462.

[2] Bellemare, M.G., Naddaf, Y., Veness, J. and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*. 47, (2013), 253–279.

[3] Chaplot, D.S., MacLellan, C., Salakhutdinov, R. and Koedinger, K. 2018. Learning Cognitive Models Using Neural Networks. *International Conference on Artificial Intelligence in Education* (2018), 43–56.

[4] Chase, W.G. and Simon, H.A. 1973. Perception in chess. *Cognitive psychology*. 4, 1 (1973), 55–81.

[5] Chi, M., VanLehn, K., Litman, D. and Jordan, P. 2011. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*. 21, 1–2 (2011), 83–113.

[6] Clark, R.E. and Estes, F. 1996. Cognitive task analysis for training. *International Journal of Educational Research*. 25, 5 (1996), 403–417.

[7] Corbett, A.T. and Anderson, J.R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*. 4, 4 (1994), 253–278.

[8] Croteau, E.A., Heffernan, N.T. and Koedinger, K.R. 2004. Why are algebra word problems difficult? Using tutorial log files and the power law of learning to select the best fitting cognitive model. *International Conference on Intelligent Tutoring Systems* (2004), 240–250.

[9] Ericsson, K.A., Prietula, M.J. and Cokely, E.T. 2007. The making of an expert. *Harvard business review*. 85, 7/8 (2007), 114.

[10] Ericsson, K.A. and Smith, J. 1991. *Toward a general theory of expertise: Prospects and limits*. Cambridge University Press.

[11] Glaser, R., Chi, M.T. and Farr, M.J. 1985. *The nature of expertise*. National Center for Research in Vocational Education Columbus, OH.

[12] Greydanus, S., Koul, A., Dodge, J. and Fern, A. 2017. Visualizing and Understanding Atari Agents. *arXiv:1711.00138 [cs]*. (Oct. 2017).

[13] Guckelsberger, C., Salge, C., Gow, J. and Cairns, P. 2017. Predicting Player Experience Without the Player.: An Exploratory Study. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, 2017), 305–315.

[14] Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[15] Kriglstein, S., Wallner, G. and Pohl, M. 2014. A User Study of Different Gameplay Visualizations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), 361–370.

[16] Kurin, V., Nowozin, S., Hofmann, K., Beyer, L. and Leibe, B. 2017. The atari grand challenge dataset. *arXiv preprint arXiv:1705.10998*. (2017).

[17] Laird, J.E., Newell, A. and Rosenbloom, P.S. 1987. Soar: An architecture for general intelligence. *Artificial intelligence*. 33, 1 (1987), 1–64.

[18] LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *nature*. 521, 7553 (2015), 436.

[19] LJPvd, M. and Hinton, G.E. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*. 9, (2008), 2579–605.

[20] Lyu, D., Yang, F., Liu, B. and Gustafson, S. 2018. SDRL: Interpretable and Data-efficient Deep Reinforcement Learning Leveraging Symbolic Planning. *arXiv:1811.00090 [cs]*. (Oct. 2018).

[21] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* (2013), 3111–3119.

[22] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K. and Ostrovski, G. 2015. Human-level control through deep reinforcement learning. *Nature*. 518, 7540 (2015), 529.

[23] Moore, S. and Stamper, J. 2019. Decision Support for an Adversarial Game Environment Using Automatic Hint Generation. *International Conference on Intelligent Tutoring Systems* (2019), 82–88.

[24] Rafferty, A.N., Brunskill, E., Griffiths, T.L. and Shafto, P. 2011. Faster teaching by POMDP planning. *International Conference on Artificial Intelligence in Education* (2011), 280–287.

[25] Stamper, J. and Barnes, T. 2009. Unsupervised MDP Value Selection for Automating ITS Capabilities. *International Working Group on Educational Data Mining*. (2009).

[26] Stamper, J., Barnes, T. and Croy, M. 2011. Enhancing the automatic generation of hints with expert seeding. *International Journal of Artificial Intelligence in Education*. 21, 1–2 (2011), 153–167.

[27] Stamper, J. and Moore, S. 2019. Exploring Teachable Humans and Teachable Agents: Human Strategies versus Agent Policies and the Basis of Expertise. *International Conference on Artificial Intelligence in Education* (2019).

[28] Such, F.P., Madhavan, V., Liu, R., Wang, R., Castro, P.S., Li, Y., Schubert, L., Bellemare, M., Clune, J. and Lehman, J. 2018. An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. *arXiv preprint arXiv:1812.07069*. (2018).

[29] Sutton, R.S. and Barto, A.G. 2018. *Reinforcement learning: An introduction*. MIT press.

[30] Van Der Maaten, L. 2014. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research*. 15, 1 (2014), 3221–3245.

[31] Verma, A., Murali, V., Singh, R., Kohli, P. and Chaudhuri, S. 2018. Programmatically interpretable reinforcement learning. *arXiv preprint arXiv:1804.02477*. (2018).

[32] Wallner, G. Play-Graph: A Methodology and Visualization Approach for the Analysis of Gameplay Data. 8.

[33] Wallner, G. and Kriglstein, S. 2012. A Spatiotemporal Visualization Approach for the Analysis of Gameplay Data. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), 1115–1124.

[34] Zahavy, T., Ben-Zrihem, N. and Mannor, S. 2016. Graying the black box: Understanding dqns. *International Conference on Machine Learning* (2016), 1899–1908.