Towers of Saliency: A Reinforcement Learning Visualization Using Immersive Environments

Nathan Douglas
University of Calgary
2500 University Dr. NW
Calgary, AB T2N 1N4 Canada
nathan.douglas@ucalgary.ca

Pablo Hernandez-Leal
Borealis AI
10020 101A Ave NW
Edmonton, AB T5J 3G2, Canada

Dianna Yim
University of Calgary
2500 University Dr. NW
Calgary, AB T2N 1N4 Canada

Frank Maurer
University of Calgary
2500 University Dr. NW
Calgary, AB T2N 1N4 Canada

Bilal Kartal
Borealis AI
10020 101A Ave NW
Edmonton, AB T5J 3G2, Canada

Matthew E. Taylor
Borealis AI
10020 101A Ave NW
Edmonton, AB T5J 3G2, Canada

ISS'19, November 10–13, 2019, Daejon, Republic of Korea.

ACM ISBN 978-1-4503-6891-9/19/11.
http://dx.doi.org/10.1145/3343055.3360747

Abstract
Deep reinforcement learning (DRL) has had many successes on complex tasks, but is typically considered a black box. Opening this black box would enable better understanding and trust of the model which can be helpful for researchers and end users to better interact with the learner. In this paper, we propose a new visualization to better analyze DRL agents and present a case study using the Pommerman benchmark domain. This visualization combines two previously proven methods for improving human understanding of systems: saliency mapping and immersive visualization.

Author Keywords
Reinforcement Learning; Data Visualization; Immersive Analytics; Virtual Reality.

CCS Concepts
•**Human-centered computing** → **Virtual reality; Heat maps;** •**Computing methodologies** → Neural networks;

Introduction
Machine learning has, in recent years, proven able to outperform humans in many different domains. However, a pervasive issue with many algorithms is that the reasoning behind their outputs can be difficult to interpret. Adadi et al. [1] describe many domains where there is an in-

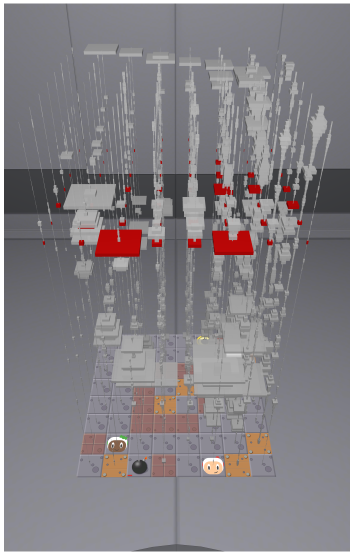

Figure 1: One tower of saliency generated by the visualization. One timestep is selected, showing the state of the board at the bottom and highlighting the saliency map for that timestep in red.

creasing demand for the implementation of more explainable artificial intelligence (XAI) such as medical, financial, and legal services. Each domain is described as needing XAI approaches to increase human acceptance and trust of the resulting decisions made by existing black-box approaches. Further, designing AI systems around humans using human-centric models can bring further benefits to the system itself by improving performance [8].

Recent work has shown immersive visualizations using head-mounted displays (HMDs), which show users fully 3-dimensional images in a virtual environment, to be a better interface for human interaction in terms of both preference and performance for some applications [4, 9]. This paper focuses on creating an explainable and immersive visualization of agents trained via deep reinforcement learning (DRL). DRL methods use deep neural networks as function approximators, which are known to generalize well to high-dimensional data at the cost of becoming black-boxes. To enhance understandability of this DRL model, our research uses a perturbation-based saliency [2] approach to identify regions of the input that are more important to the corresponding output. This abstracted metadata about the input can be visualized rather than visualizing the underlying model itself. This paper demonstrates the proposed visualization system by using data generated by a DRL agent trained using the A3C algorithm [5] in the recent multiagent benchmark of Pommerman [7].

Related Works

Other works have explored visualizing DRL agents in similar domains but, to the best of our knowledge, no other systems exist that visualize agents using immersive technologies. The DQNViz system created by Wang et al. [10] contains many different charts and graphs, each tailored to different data types and scales. All of these graphics are very standard bar, line, and pie charts presenting information about the DQN model [6], but require expertise in DRL to interpret.

Greydanus et al. [2] used the perturbation-based saliency method to determine and visualize important regions of an agent's input. Their results were shown to improve laymen's interpretation of the agent's reasoning. However, the amount of information their visualization could show was limited by being implemented in 2-dimensions. The benefit of using immersive technologies is that it provides an extra dimension in space to assist in user understanding given spatial context, as well as removing the space limitation with computer screens. An example of this is a tool developed by Usher et al. [9] that increased the productivity of neuroscientists tasked with tracing neurons in brain scans.

Towers of Saliency

This section describes the two main components that work together to create the visualization: (1) the saliency generation method, which generates abstracted data from the learner to visualize, (2) the Towers of Saliency immersive environment, which visualizes the resulting saliency data.

Saliency Generation

In general, saliency mapping techniques generate heatmaps of important areas in a given input. These important areas are called more salient. Traditional saliency algorithms [3] calculate salient areas based on properties of the input (e.g., changes in pixel color and intensity). However, we use a perturbation-based technique that measures how the agent reacts to small changes in the input, thus the salient areas are based on the agent's interpretation of the image. This abstracted metadata about the agent can then be used to visualize the agent's behaviour.

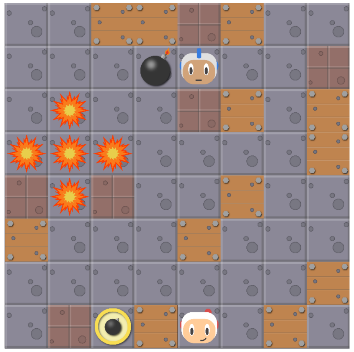

Figure 2: A still of the Pommerman game. The game is played in an 8 by 8 tile grid. Players can take one of six actions in each timestep: Up, Down, Left, Right, Pass, and Place a bomb.

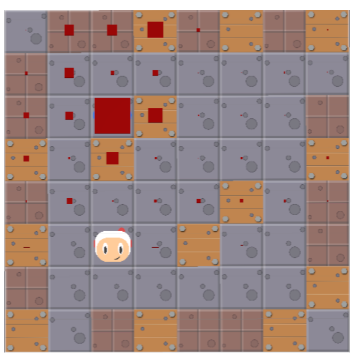

Figure 3: A top down view of a board with one heatmap overlaid. A larger red box over a tile indicates a greater saliency value.

Most saliency mapping implementations expect images as input, measuring or modifying the pixels to calculate the salient areas. However, the agent used in this case takes a matrix recreation of the game board as input rather than an image, while still using the agent's policy, a probability distribution over all possible actions, as output. As such, this system had to alter the algorithm used by Greydanus et al. [2] to modify tiles rather than groups of pixels. For a given input, our algorithm first changes the categorical value of one index in the matrix to a value the agent has never seen before. This modified input is given to the agent, and the change in the agent's output is measured. This tile altering method is repeated for each tile on the board, creating a mapping of which tiles, when changed, most affected the agent's output.

Another modification made in our algorithm is that the changes in output are recorded for each action, rather than computing the squared magnitude of the changes in all actions. Thus, several saliency maps can be generated with respect to each action the agent can take. However, the change in an action can be in the range $[-1, 1]$, so the absolute value of each change is used to normalize values in the range $[0, 1]$. This emphasizes changes in general, at the cost of distinguishing positive and negative changes.

Towers of Saliency

The Towers of Saliency visualization aims to provide information about the agent's actions over an entire game. A game consists of a series of consecutive, discrete timesteps that finishes when the game is won or lost. In every step, both the state and the saliency data are generated and stored to be visualized later.

The visualization itself uses all of the data recorded from a given game to create the towers. At the bottom of the tower is a recreation of the game board that functions as the foundation of the tower. On top of the board, a saliency map is generated by creating a plane of rectangular prisms, one prism for each tile on the game board. Each prism has a constant height and a square base scaled to equate to the saliency value for the tile underneath it. Each tile has a saliency value in the range $[0, 1]$ and is thus used as a linear scale of the prism's dimensions. This saliency map creation process is then repeated for each state in the game, each creating another layer of prisms immediately above the last. Additionally, a single state can be selected which highlights the saliency map for that state. The board at the bottom of the tower also changes to represent the environment during that state.

Structured this way, a tower can provide a large amount of information to the user. Seeing multiple maps allows users to identify what the agent identifies as important areas of the game's environment and also compare how those areas change over time. Further, since saliency data is recorded with respect to each action the agent can take, multiple towers can be generated using the saliency data of different actions.

Users view and interact with these towers using any OpenVR HMD and controllers. Within the immersive environment users can walk around the towers, each of which have an approximate size of one square meter wide and two meters tall. Each tower can be dragged and placed within the environment to better allow users to view the visualization in their preferred format.

Conclusion and Future Work

This system is an immersive tool to analyze abstracted information about a DRL agent. This system leaves many areas to further explore. First, the raw saliency values produced by our saliency algorithm are in the range $[-1, 1]$, but the current visualization displays the absolute value of each tile. In the future, we aim to display negative and positive values using different visual attributes. Also, the Pommerman environment used in this system operates on fixed, discrete timesteps. This allows this system to create planes that are all of an equal height. Another similar system that uses either discrete timesteps of varying length or a continuous time axis would need to consider how to appropriately control the width of each saliency map displayed.

REFERENCES

[1] A. Adadi and M. Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. DOI: http://dx.doi.org/10.1109/ACCESS.2018.2870052

[2] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. 2017. Visualizing and Understanding Atari Agents. *CoRR* abs/1711.00138 (2017). http://arxiv.org/abs/1711.00138

[3] L. Itti, C. Koch, and E. Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 11 (Nov 1998), 1254–1259. DOI:http://dx.doi.org/10.1109/34.730558

[4] O. Kwon, C. Muelder, K. Lee, and K. Ma. 2016. A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics* 22, 7 (July 2016), 1802–1815. DOI: http://dx.doi.org/10.1109/TVCG.2016.2520921

[5] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. *CoRR* abs/1602.01783 (2016). http://arxiv.org/abs/1602.01783

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (25 Feb 2015), 529 EP –. https://doi.org/10.1038/nature14236

[7] Cinjon Resnick, Wes Eldridge, David Ha, Denny Britz, Jakob Foerster, Julian Togelius, Kyunghyun Cho, and Joan Bruna. 2018. Pommerman: A Multi-Agent Playground. *CoRR* abs/1809.07124 (2018). http://arxiv.org/abs/1809.07124

[8] G. K. L. Tam, V. Kothari, and M. Chen. 2017. An Analysis of Machine- and Human-Analytics in Classification. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 71–80. DOI: http://dx.doi.org/10.1109/TVCG.2016.2598829

[9] W. Usher, P. Klacansky, F. Federer, P. Bremer, A. Knoll, J. Yarch, A. Angelucci, and V. Pascucci. 2018. A Virtual Reality Visualization Tool for Neuron Tracing. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 994–1003. DOI: http://dx.doi.org/10.1109/TVCG.2017.2744079

[10] J. Wang, L. Gou, H. Shen, and H. Yang. 2019. DQNViz: A Visual Analytics Approach to Understand Deep Q-Networks. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (Jan 2019), 288–298. DOI: http://dx.doi.org/10.1109/TVCG.2018.2864504