A Literature Review of Machine Learning in AI Interaction with Human-centric Interfaces for novel tasks such as Tetris

Tobias Dean
Earlham College
Richmond, Indiana
tadean21@earlham.edu

ABSTRACT

This document is a brief literature review of articles and documents covering the uses of Machine Learning to train an AI to interact with user interfaces intended for humans. In particular, this literature review focuses on how humans play Tetris and how to best train an AI to play Tetris in a similar manner with the same information, that is to say, the current screen and access to all the controls. This article is divided into several sections introducing the concept, explaining the focus on Tetris, and the mechanics behind the project. This document was expressly written for CS388 at Earlham College.

CCS CONCEPTS

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

KEYWORDS

Machine Learning, AI, Tetris, General video game playing, Video Game Description Language

ACM Reference Format:

Tobias Dean. 2018. A Literature Review of Machine Learning in AI Interaction with Human-centric Interfaces for novel tasks such as Tetris. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

One of the long-standing goals of research on Machine Learning and Artificial Intelligence in general is to develop AI that displays general intelligence which allows for the AI to fulfil multiple roles. [2] Ideally, such an AI would also be versatile enough to complete tasks using pre-existing interfaces designed for human use. This would be an obvious advantage as the developers looking to use an AI with general intelligence would not always have the luxury of changing the interface or AI to allow the latter to interpret the former easily. [2]

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Therefore, designing an AI that can gather information and complete a task by simply looking at a computer's screen and knowing what keys it can access is an important step toward developing a versatile AI that can operate at the same level as a human. Such an important step can not be taken all at once, and thus, a smaller benchmark must first be met. A suitable first step would be to develop an AI capable of playing Tetris with only knowledge of what the current screen looks like, what keys it can send the signal of pressing, and a goal to increase the score as much as possible. Fortunately, Tetris and video games as a whole have long served as a testing ground for machine learning and AI, and there is no shortage of foundational research on this topic.

2 WHY TETRIS?

There are a number of reasons Tetris stands out as an appealing challenge for this avenue of research. The first and most simple reason is that Tetris is an old, well-known, and simple game. For this reason, Tetris has a long history of use in testing machine learning. The article *The game of Tetris in machine learning* [1] by Algorta and Şimşek details how Tetris has been used as a testing ground for machine learning AI as early as 1996. This same article also covers how much success different approaches have had over time. The earliest attempt discussed simply used large-scale feature-based dynamic programming and tracked the number of holes and the height of the highest column. Later attempts would go on to include hand-crafted agents, genetic algorithms, and approximate modified policy iteration. [1]

Furthermore, the article details how, over time, certain practices, such as using the scoring system of Tetris' original implementation over its more complicated successors, have become the standard for testing AI within Tetris. It is important to consider which implementation of Tetris one uses in these tests, as some versions award more points for clearing several rows at a time. Additionally, implementations vary on whether or not certain moves, like sliding a piece under a placed piece with a gap underneath, a move which is known as an overhang are allowed. [1] Differences between implementations of the game must be considered when comparing AI performances, as having one AI having access to more possible moves or getting more points for completing the same number of rows can make it look unfairly superior to an AI tested in an implementation featuring neither of those features.

Naturally, such a long-used benchmark has a wellspring of resources and foundational research tied to it, such as *The game of Tetris in machine learning* [1] and *Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris* [5], making it all the more appealing to use. Tetris is, however, not the only long-standing benchmark, and its veterancy as a testing ground is not the end of its merits. Tetris also offers a simple yet sufficiently complex challenge that features a random element that will prevent the AI from simply learning an optimal series of keys to press without being able to adapt this learned method to a task where even one detail is different from the training set.

For those reasons and a moderate personal familiarity with the game, Tetris was settled upon as the testing environment for this research project.

3 HOW HUMANS PLAY TETRIS

When making an AI with the goal of interacting with its tasks in a very human-like manner, it is prudent to know how a human interacts with their tasks. In this case, *Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris*, written by Jermann et al. sheds some light on the mystery. The article covers the findings of a study on eye movement during collaborative work conducted on subjects playing a game of Tetris modified to be played by two players working together. Most interestingly, this study's focus on the differences and interactions between "good" and "bad" players' eye movements gives some insight into how to direct the AI's proverbial eyesight. The article notes how more proficient players trended towards "lateral gaze sequences" than the less experienced players. Additionally, the article also breaks down what the key parts of a Tetris game screen are, specifically the player's tetromino, their game piece, and the contour of the current board created by already placed pieces. [5]

Naturally, a wealth of tips and advice for how to expertly play Tetris are also available, as one would expect with such an old and well-known game. In an interview with Jonas Neubauer, a seven-time winner of the Classic Tetris World Championship, he provided a few strategies for playing Tetris. [7] When placing pieces, he recommended deliberately keeping the field as even as possible and, when necessary, stacking unevenly in the middle of the grid. Additionally, he recommends holding an I tetromino to clear four rows at once and memorising which colour corresponds to which shape in order to save time checking which piece is next. Although some strategies for humans, such as Neubauer's suggestion of practising with Tetris games running at higher speeds to improve reflexes, may not directly translate to AI, many of his suggestions are well worth integrating where possible.

Between these guides and the aforementioned article, *Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris*, we can establish a clear baseline for what an optimal, or at the very least acceptable, human interpretation of the game screen looks like. Using this baseline, we can guide our development of a suitable machine learning AI much more efficiently.

4 AI IMPLEMENTATIONS

As previously mentioned, there is already a significant number of well-documented machine learning AI capable of playing Tetris. Therefore, there will be no shortage of previous works to draw inspiration from. Keeping the goal of versatility in mind, it is worth studying the AI entered into the general game playing competition featured in the intuitively named article, *General video game playing* [2]. The AIs built for that competition were centred around versatility and having no previous experience with a situation.

One of the strategies used by previous attempts was the implementation of evolutionary algorithms. [4] Looking at their report on this study, we can see several interesting choices the authors made, which may inform our own choices. This implementation chooses the best move by assessing how desirable potential subsequent boards are based on several sub-ratings, the weights of which are determined by the genetic algorithm. Additionally, this implementation chooses to use a heuristic approach when rating boards in order to act more quickly, a valid concern as Tetris pieces continue to fall from the moment they enter the board. The board quality is assessed based on 12 metrics, but the authors also acknowledge that using more complicated calculations to rate how desirable a board is can have advantages and disadvantages. Using more criteria to judge a board led to better performance but made that AI more niche for the exact environment it was trained in. [4] Although not every aspect of this AI implementation is in line with this study's goals of a human-like perspective, such as using the number of pieces placed to asses AI performance, it is still very worth building on these studies' work rather than starting from scratch.

Another team's work that is rather interesting and, more importantly, relevant to this project is that which is documented in the paper *Tetris Artificial Intelligence*. [9] This team built a Tetris AI that, like other agents already developed, saves time by choosing not to calculate every possible game state a piece can create. However, of particular interest is their decision to make three kinds of AI, each using different strategies. The first and most obvious strategy was used by the Greedy AI, which prioritised getting a line clear whenever possible. A bit more sophisticated was the Tetris AI, which waits until it can score a 4-row elimination, also known as scoring a Tetris, by waiting until it has an I piece and four-row hole to fill. Although both of these AI cleared about the same number of rows, the Tetris AI got much higher scores due to the bonus points clearing more than one line at a time grants. Lastly and most successful was the Two Wide Combo AI, which focuses not on how many points it can get with a single action but on getting combo eliminations. The final human strategy adapted into their AI proved to be the most effective at reaching a high score.

5 SUITABLE TESTING ENVIRONMENTS

Another essential aspect to consider for this project is the specific details of the environment in which the study will be conducted. Although the base game of Tetris seems the obvious choice using a standard game for testing creates some complications. When covering aspects of testing in Tetris that have become standard *The game of Tetris in machine learning* points out how the need for faster testing caused changes like reducing the game size to 10 by 10 from 20 by 10. This change shortens the game, allowing more games to be completed in a shorter time. With this in mind, it becomes clear why an implementation of Tetris that can be modified on the fly is important.

Conveniently, the general game playing competition has led to the development of a uniquely qualified environment. The companion article to *General video game playing* [2], *Towards a Video Game Description Language* [3] goes into depth about the need for a programming language designed to be easy to interpret for humans and AI alike. VGDL was written with features that make games implemented in it conducive to quick modifications and easy interactions with AI made for general gameplaying. For these reasons, this project will conduct all of its testing in an implementation of Tetris written in VGDL.

There are a number of articles written subsequent to *Towards a Video Game Description Language* [3] that describe the author's implementations of VGDL. One such article, *XML-Based Video Game Description Language* [6], written by Quiñones and Fernádez-Leiva describes a more recent implementation of VGDL implemented using XML that the authors published to github. Naturally, the article also explains how to use this implementation and compares it to other versions of VGDL.

Just as naturally, XVGDL is not the only available implementation of VGDL, and another article *A Video Game Description Language for Model-based or Interactive Learning* [8] details an implementation of VGDL in PyGame. This other implementation, PyVGDL, shows how an implementation of VGDL tailored to more specific needs is far from hard to find.

6 CONCLUSION

Reviewing the already published works in the field it is easy to see the path laid out. There is already an extensive history of Tetris as a benchmark for machine learning tests and an even more extensive history of humans playing Tetris to compare it to. The groundwork for where and how to implement suitable AI has already begun in force.

It only seems natural then that the next milestone in making such AI more versatile is finding the process of making AI that can work with the same limited information a human has when working with the same task. Although that is a broad goal a feasible step towards that goal is to get an AI to learn to play Tetris the same way a human unfamiliar with the game would learn.

ACKNOWLEDGMENTS

To David, for teaching me how to write a Literature Review and helping to refine my work in CS388.

REFERENCES

[1] Simon Algorta and Özgür Şimşek. 2019. The Game of Tetris in Machine Learning. (May 2019), 1–7. https://arxiv.org/abs/1905.01652 arXiv:1905.01652v2 [cs.LG].

[2] Marc Ebner-Graham Kendall Simon M. Lucas Risto Miikkulainen Tom Schaul Tommy Thompson Michael Mateas Mike Preuss Pieter Spronck John Levine, Clare Bates Congdon and Julian Togelius. 2013. General Video Game Playing. 6 (2013), 1–7. https://strathprints.strath.ac.uk/57217/ General video game playing. Artificial and Computational Intelligence in Games. Dagstuhl Follow-Ups, Volume 6, ISBN 978-3-939897-62-0.,,; pp. 77–83 Dagstuhl Publishing Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany.

[3] Simon M. Lucas-Tom Schaul Tommy Thompson Julian Togelius Michael Mateas Mike Preuss Pieter Spronck Marc Ebner, John Levine and Julian Togelius. 2013. Towards a Video Game Description Language. 6 (2013), 1–16. https://strathprints.strath.ac.uk/45278/ Artificial and Computational Intelligence in Games. Dagstuhl Follow-Ups, Volume 6, ISBN 978-3-939897-62-0.; pp. 85–100 Dagstuhl Publishing Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany.

[4] Gabriella Kókai Niko Bhm and Stefan Mandl. 2005. An Evolutionary Approach to Tetris. (2005). https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b0fe1ed14404db2eb1db6a777961440723d6e06f

[5] Marc-Antoine Nüssli Patrick Jermann and Weifeng Li. 2010. Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris. (Sept. 2010), 1–9. https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/HCI2010.7

[6] Jorge R. Quiñones and Antonio J. Fernández-Leiva. 2019. XML-Based Video Game Description Language. *IEEE Access* 8 (2019), 4679–4692. https://doi.org/10.1109/ACCESS.2019.2962969

[7] Jeff Ramos. 2019. Tetris tips from a seven-time world champion; How to begin mastering the classic puzzle game. *Polygon* (2019). https://www.polygon.com/guides/2019/2/22/18225349/tetris-strategy-tips-how-to-jonas-neubauer

[8] Tom Schaul. 2013. A Video Game Description Language for Model-based or Interactive Learning. 6 (2013), 1–8. https://strathprints.strath.ac.uk/45278/ Artificial and Computational Intelligence in Games. Dagstuhl Follow-Ups, 6 . Dagstuhl Publishing, Wadern, pp. 85-100. ISBN 9783939897620..

[9] Wei-Chiu Ma Tian-Li Yu Wei-Tze Tsai, Chi-Hsien Yen. 2012. Tetris Artificial Intelligence. *NTUEE, Taiwan* (2012). https://chihsienyen.github.io/pdf/Tetris_AI.pdf

Received 9 October 2023