challenges

MDPI

Perspective

Considerations for Comparing Video Game AI Agents with Humans

Christopher R. Madan

School of Psychology, University of Nottingham, Nottingham NG7 2RD, UK;
christopher.madan@nottingham.ac.uk

Received: 25 July 2020; Accepted: 17 August 2020; Published: 20 August 2020

Abstract: Video games are sometimes used as environments to evaluate AI agents' ability to develop and execute complex action sequences to maximize a defined reward. However, humans cannot match the fine precision of the timed actions of AI agents; in games such as StarCraft, build orders take the place of chess opening gambits. However, unlike strategy games, such as chess and Go, video games also rely heavily on sensorimotor precision. If the "finding" was merely that AI agents have superhuman reaction times and precision, none would be surprised. The goal is rather to look at adaptive reasoning and strategies produced by AI agents that may replicate human approaches or even result in strategies not previously produced by humans. Here, I will provide: (1) an overview of observations where AI agents are perhaps not being fairly evaluated relative to humans, (2) a potential approach for making this comparison more appropriate, and (3) highlight some important recent advances in video game play provided by AI agents.

Keywords: AlphaStar; explainable AI; StarCraft; Rubik's Cube; AlphaGo; machine learning; reinforcement learning; robot psychology

1. Introduction

Video games are sometimes used as test, or "gym", environments for evaluating AI approaches to learning and acting [1–6]. Within these games, AI agents are sometimes compared to human performance as a benchmark [4,7,8].

Often, many of these studies use a variety of retro games, such as for Atari 2600 (Atari Inc., Sunnyvale, CA, USA) or the original Nintendo Entertainment System (NES) (Nintendo Co., Ltd., Kyoto, Japan), to demonstrate the generalisability of an AI agent to play many games and optimise actions based on minimal coding that is unique to each game, such as to know where the score is shown (see Figure 1A). Some games are now well known to be difficult for AI agents, such as the Atari 2600 games Montezuma's Revenge (Parker Brothers, Beverly, MA, USA) and Battle Zone (Atari Inc., Sunnyvale, CA, USA), where the mapping between increases in calculable score and actions are particularly complex, with subgoals not clearly defined [9,10]. Other recent advances in AI agents have been in strategy games that involve direct competition, such as chess and Go. More recently, Vinyals et al. [8] developed AlphaStar to play the competitive computer-based strategy game StarCraft 2 (Blizzard Entertainment, Inc., Irvine, CA, USA). The original StarCraft (Blizzard Entertainment, Inc., Irvine, CA, USA) computer game has been a staple in AI research for well over a decade [11–13], but with the recent advances in both deep learning and reinforcement learning, the successor game of StarCraft 2 has stood as a clear benchmark for future developments in video game AI.

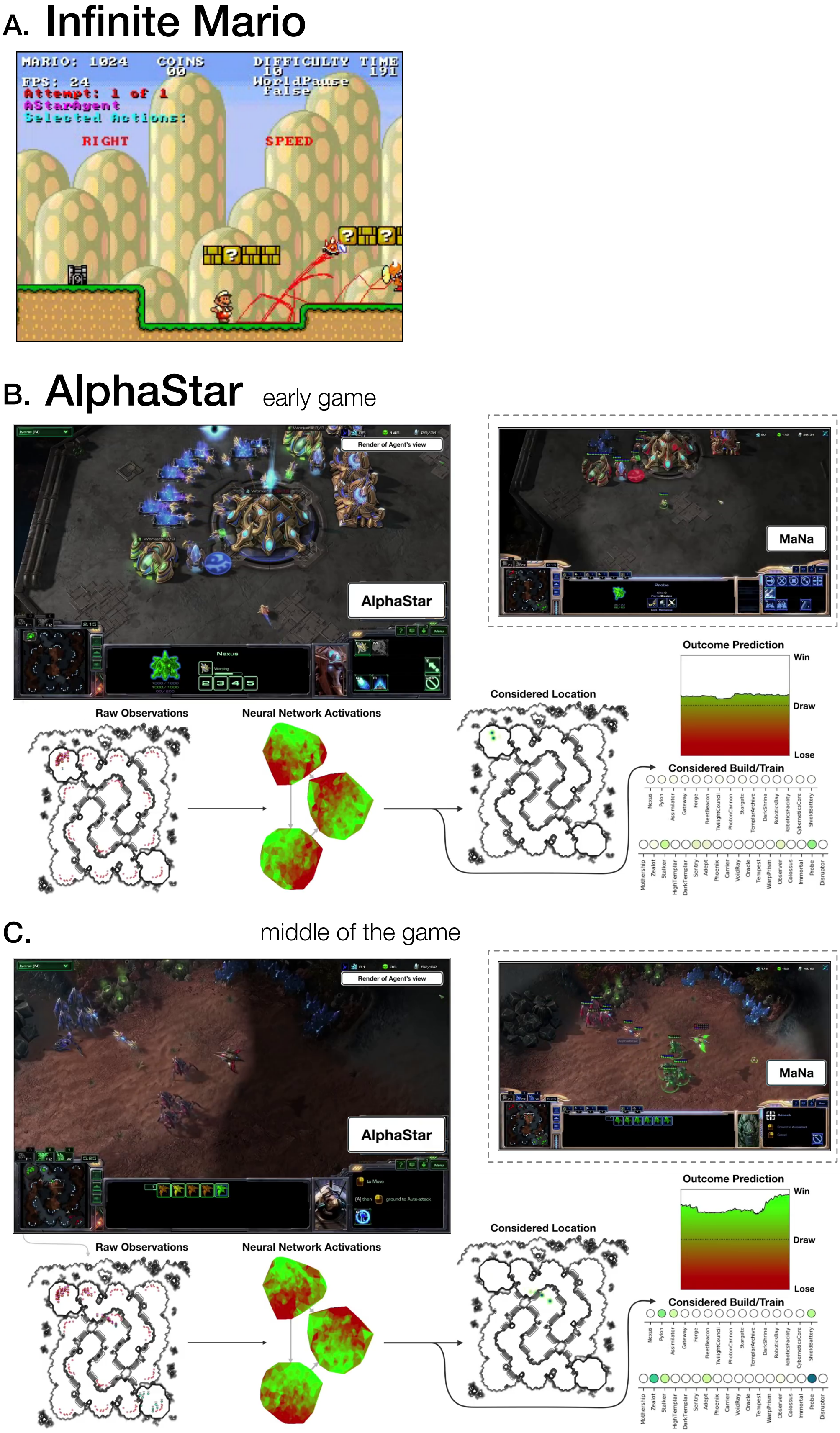

Figure 1. Agent visualisation for (**A**) Infinite Mario and AlphaStar in the (**B**) early and (**C**) middle of the game. Panels adapted from [14,15], respectively.

Briefly, StarCraft and StarCraft 2 are real-time strategy (RTS) games where each player is a commander of a science-fiction military force of one of three species: Terran (advanced humans), Protoss (advanced technological aliens), or Zerg (advanced biological aliens). From this role, you are tasked with building one or more military bases, harvesting resources from around the base, and building any of a variety of military units. The goal of each game is to defeat the other players (i.e., destroy their bases). Each species involves different strategies and has approximately 15 unique unit types available to it. Units vary in characteristics such as attack range, movement, which is either ground-based or flying, and abilities such as being able to gather resources, construct buildings, carry other units, heal/repair other units, or even limited invisibility. All species have units that are designed to be suitable counters to units from each species, reflecting a rock-paper-scissors dynamic.

AlphaStar has been provided with a direct API to the StarCraft game engine, allowing the agent to skip past the tedious and arguably less interesting problems of computer vision and sensorimotor precision (e.g., if this was an independent robot that had to control a computer keyboard and mouse); see Figure 1B,C. Early versions of the AlphaStar agent did not need to control the camera and had an all-seeing perspective of all actively viewable regions, as well as inhuman "micro" precision of actions. Thus, while AlphaStar was able to develop novel and intriguing strategies, this is not its only advantage relative to a human player. In this article, I will discuss considerations for how AI agents can be made more comparable to humans, allowing us to learn the most from their strategies, rather than their capabilities that we cannot hope to match.

Using AlphaStar [8] as the main example, one approach to adjust for this is to limit AI agents to the same abilities as expert humans:

> Camera view. Humans play StarCraft through a screen that displays only part of the map along with a high-level view of the entire map (to avoid information overload, for example). The agent interacts with the game through a similar camera-like interface, which naturally imposes an economy of attention, so that the agent chooses which area it fully sees and interacts with. The agent can move the camera as an action. [. . .]
>
> APM limits. Humans are physically limited in the number of actions per minute (APM) they can execute. Our agent has a monitoring layer that enforces APM limitations. This introduces an action economy that requires actions to be prioritized. [. . .] [A]gent actions are hard to compare with human actions (computers can precisely execute different actions from step to step).

Indeed, it was generally agreed that "forcing AlphaStar to use a camera helped level the playing field" [16]. As such, even with the camera limitations later added to AlphaStar [8], the agent is able to click on objects on the screen that are not actually visible to human experts (i.e., objects at the end of the screen with only a few viewable pixels) ([17] confirmed as replay "AlphaStarMid_053_TvZ"). Even with these constraints, it can be debated whether the limitations imposed on AlphaStar are sufficient given that humans have considerable limitations in their sensory and motor abilities, for instance: (1) the inability to actively attend to all of the presented visual information simultaneously; and (2) less targeting control of the temporal and spatial precision of their actions. In human experts, declines in performance can emerge as early as at 24 years old [18]. For instance, even though humans and AlphaStar could be matched to the same APM distribution characteristics, human actions are not all efficiently made—unlike AlphaStar. In a preliminary show-match [19,20], AlphaStar was able to demonstrate superhuman control, "it could attack with a big group of Stalkers, have the front row of stalkers take some damage, and then blink them to the rear of the army before they got killed" [16] (watch the demonstration with commentary [20] from 1:30:15). Humans have previously used this strategy as well, but are unable to have the spatial and temporal precision to execute it as well as AlphaStar is able. In another example, AlphaStar can exhibit precision control of multiple groups of units beyond human capabilities (e.g., see 1:41:35 and 1:43:30 in the previous video [20]). Clearly, human performance, even that of skilled humans, is not a suitable benchmark for modern AI agents;

they simply are not matched in the sensorimotor processing and precision necessary for comparable real-time performance. Note, however, I am not critiquing the differences in the amount of experience between humans and AI agents, e.g., "During training, each agent experienced up to 200 years of real-time StarCraft play". While AI agents often are provided with orders of magnitude more experience than human agents in the tested scenario, humans benefit from a biological architecture that makes this relative difference hard to evaluate directly (e.g., genetic and physiological optimisation; see Zador [21] and LeDoux [22]). Moreover, I suggest that the principle measure of interest is not how quickly an agent reaches its current level of performance (e.g., number of matches or years of play), but the decision-making strategies that were present at the point of evaluation. Notably, these highlighted issues do not apply to turn-based strategy games, such as chess or Go, only to real-time strategy games. Moreover, it is also important to acknowledge that these concerns about AlphaStar's lack of limitations (e.g., camera view and APM limits) were not as relevant to previous StarCraft AI agents, which were not nearly as able to defeat human players [23].

Issues in comparing AI agents to humans are, of course, not limited to AlphaStar. In work using retro games where humans have been included as a benchmark, these may not truly be performance from expert humans. Mnih et al. [7] reported many measures based on comparisons to a "professional human games tester"; however, this appears to be a single human for all games and likely was not an expert in any of the games. The level of expertise for the human data here is particularly questionable, for instance the human score for Seaquest was 20,182 points (see Extended Data Table 2 of [7]), whereas a community high-score website with rigorous reporting procedures has 276,510 as the current world high-score [24]—a value 10 times higher than the expert in Mnih et al. [7]. Moreover, using tool-assistance (described in more detail later on), humans have achieved the maximum score possible—999,999 points [25]. These tool-assisted playthroughs can be precisely re-played based on a recorded set of button presses with frame-by-frame precision. This is not unique to only a single example, with other high scores being also well above those obtained by the human in Mnih et al. [7] (e.g., Kangaroo, 3035 vs. 47,800 [26]). Similar critiques have also been raised by Toromanoff et al. [27]. Even in more recent work (e.g., [28–30]), obtaining appropriate human benchmarks when making claims of the AI agent "achiev[ing] superhuman performance in a range of challenging and visually complex domains" [30] appears to be an afterthought.

2. Learning from the Cube

While the goal of this paper is to consider how machine-learning algorithms can be fairly compared to humans using video game benchmarks, a simplified case is also worth considering—the Rubik's Cube—which itself is no stranger to machine-learning optimisation (e.g., [31–33]). The Rubik's Cube, which arguably could be implemented as a video game itself, is a well-known game where the goal state is a $3 \times 3 \times 3$ cube where each face of the cube is only a single colour, and otherwise is comprised of six unique colours. The initialisation state is any valid configuration of colours that can be obtained from an indeterminate number of twists such that the cube is considered scrambled. As such, the state space of the Rubik's Cube is 4.33×10^{19} possible configurations (i.e., states). It is worth considering that the state space of a Rubik's Cube is significantly smaller than any permutation of the six colours along the 9×6 cubies (i.e., the visible small square panels that make up each face), as for instance, the centre white cubie is always directly opposite the centre yellow cubie. For every valid Rubik's Cube state, there are 11 unreachable states based on how colours along the cubies are inter-related; in other words, if one were to randomise the coloured stickers on a Rubik's Cube, there is a $11/12$ (92%) chance that it would be unsolvable.

The so-called "beginner's method" for solving the Rubik's Cube [34] often averages more than 100 moves to solve the cube, where each step corresponds to a quarter-turn of a face or middle, either clockwise or counter-clockwise—or a half turn, or a rotation of the cube's perspective. Here, the methods are comprised of numerous algorithms, i.e., sequences of moves, designed to swap colours from portions of the cube or otherwise reach intermediate goal states (e.g., cross/"daisy", complete layer, 2 × 3 × 1 block), as illustrated in Figure 2.

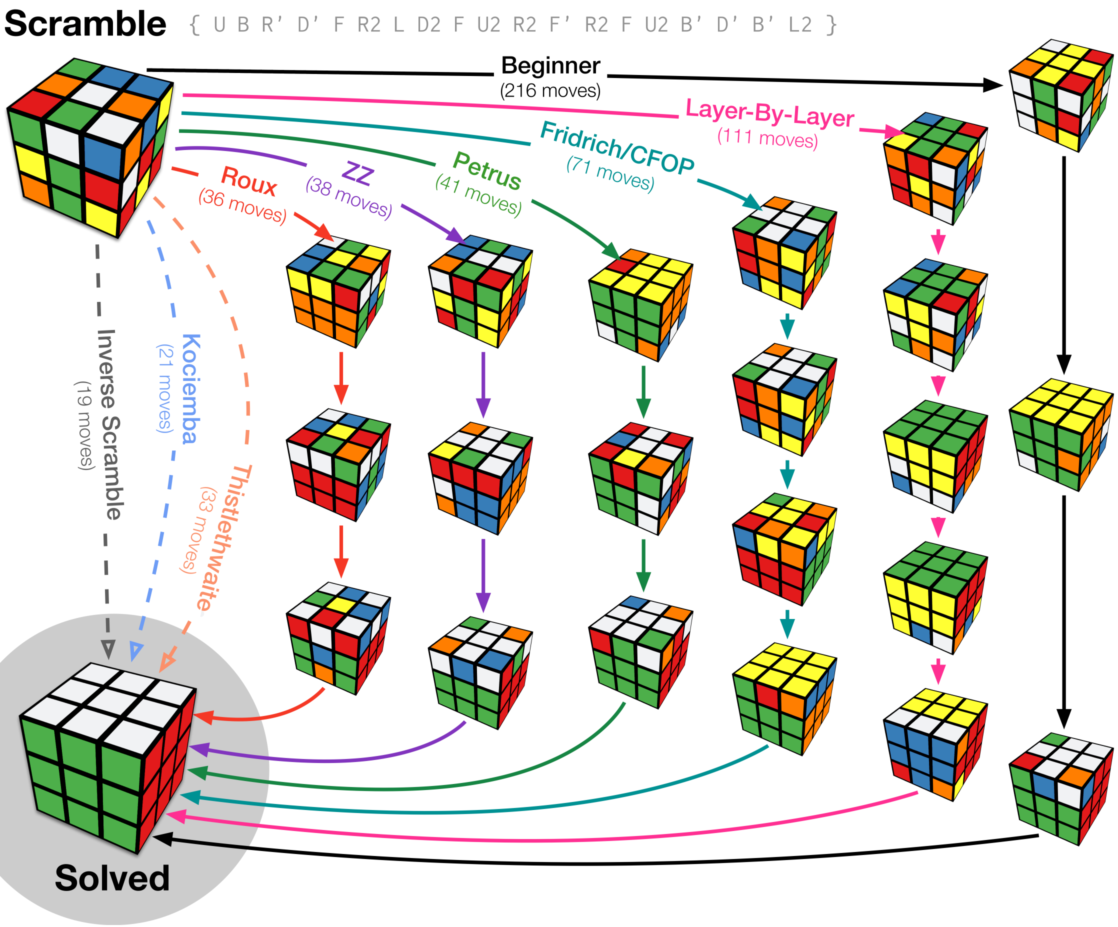

Figure 2. Illustration of various methods for solving a scrambled Rubik's Cube. A scrambled cube, with the scramble notation shown at the top, can be solved using a variety of methods. The number of moves required to solve the cube using each method is shown, along with pictures of intermediate goal states (e.g., cross/"daisy", complete layer, 2 × 3 × 1 block).

Competitive approaches for solving the cube rely on increasing numbers of algorithms and increasing complexity of the intermediate goal states [35]. For instance, the beginner layer-by-layer method only has a few defined/memorised algorithms, but a popular competitive solving method, the Friedrich/CFOP method, has 78 to 119 algorithms (depending on the variant of the method) and averages 55 moves to solve the cube. The ZZ method relies on more complicated intermediate goal states, as well as more memorised algorithms (up to 537), but results in slightly fewer moves, averaging in the low 40s. Most competitive Rubik's Cube solvers, known as "cubers", use the CFOP method due to its balance between the number of algorithms to memorise and the number of moves, even though the Roux and ZZ methods would reduce the number of necessary moves to solve the cube; most competitive cubing is based on overall solving speed, not number of moves, which are only indirectly related. As such, here, a limitation for humans even within tractable strategies is the algorithmic complexity to identify and execute more efficient solving strategies, even when this can be dissociated from the time related to sensorimotor actions.

Thistlewaite [36] proposed a method that could solve a Rubik's Cube from any initialisation state in up to 52 moves, later refined to 45 moves. However, this approach requires too many algorithms for a human to memorise and functions more as a look-up table of input and output state configurations, although it still involves intermediate goal states. Kociemba refined this approach by reducing the number of intermediate goal states, reducing the number of steps required to solve any cube to up to 29 moves. Later work has shown that any possible configuration of a Rubik's Cube can be transformed to the goal state using no more than 20 moves [37], and from this, a complete look-up table of all 4.33×10^{19} valid states could be stored along with their associated solving moves, fully replacing the strategy with pre-memorised optimal solutions. While I do not think it is necessary for AI agents to be compared to humans, some researchers have developed robots that can solve a physical Rubik's Cube, rather than merely virtual representations of it (e.g., [38,39]).

3. Developing Better Benchmarks for AI Agents

A better comparison for real-time AI agents would be tool-assisted human performance, where the precise timing of actions is optimised frame-by-frame, as is done in tool-assisted speed-runs (TAS) [40]. In this approach, humans can play a game with the option of slowing it down and timing their sequence of button presses to perform actions with frame-by-frame temporal precision (as retro games do not use a mouse, the spatial precision concern with AlphaStar does not apply here). This approach would remove the definitively superhuman sensorimotor advantage of AI agents, while still allowing for the comparison of strategic planning abilities. This level of fine control allows for a theoretically perfect playthrough that would otherwise be nearly impossible to perform by a human. The usual goal of this approach is a "speed-run", that is to play through the game as quickly as possible, but could be readily applied by maximising a score or any other goal criteria. This would provide the humans with external aids to level the playing field and focus more on the strategic component of the gym environment (also see [41]). Some tool-assisted speed-runs use modified hardware to directly provide inputs into the original video game console as a controller [42]; however, in most cases, speed-runs are evaluated using software emulation [43].

4. AI Agents Performing beyond Human Capabilities

The suggestions here are not intended to be overly critical, just more "fair" of a comparison. With the Rubik's Cube, advanced strategies—i.e., those beyond the capabilities of humans—can transition from the initialisation state to being solved without requiring the more comprehensible intermediate states, such as a white daisy/cross as a first step. As outlined earlier, any Rubik's Cube state can be solved to the goal state in twenty moves or less [37]. This same work identified the number of actions required to solve a cube from each of the possible 4.33×10^{19} initialisation states. Based on this algorithm, only 3% of states can be solved in 16 moves or less, but this increases to 30% for 17 moves or less and up to 97% for 18 moves or less. In contrast, common human strategies will often take at least 40 to 60 moves, though as highlighted earlier, these strategies are developed to optimise human processing and execution time (i.e., speed-cubing), not minimising moves taken. The algorithmic complexity of planning the optimal Rubik's Cube state a dozen moves ahead, without executing them yet, is not a task humans are well suited for, which is why the human solving methods involve memorising algorithms that help achieve intermediate goal states, somewhat similar to chess gambits and Go joseki.

Of interest are surprising examples of emergent behaviour. Chrabaszcz et al. [44] found that AI agents have the potential to grossly out-perform humans and find novel solutions, even if an "environment" has been available to humans for decades. In their work with a Q*bert agent, "the agent learns that it can jump off the platform when the enemy is right next to it, because the enemy will follow: although the agent loses a life, killing the enemy yields enough points to gain an extra life again. The agent repeats this cycle of suicide and killing the opponent over and over again". More impressively, however, "the agent discovers an in-game bug. First, it completes the first level and then starts to jump from platform to platform in what seems to be a random manner. For a reason unknown to us, the game does not advance to the second round but the platforms start to blink and the agent quickly gains a huge amount of points (close to 1 million for our episode time limit)". This behaviour has since been reproduced in the console version of Q*bert by a human [45], ruling out the possibility that this behaviour was only possible (1) due to a bug in the game emulation or (2) relying on rapid button presses that exceeded plausible human reaction times. In another example, and more comically, reference [46] reported on an overly greedy AI agent that can play Tetris—but not well—placing blocks haphazardly for their associated increase in points, but learning to pause the game just before the next block being generated would cause the game to be over. See [47] for further examples of emergent behaviour.

In a recent study, OpenAI researchers have constructed an environment where multiple AI agents play a game of hide-and-seek in teams and adapt to each others' strategies, in a form of multi-agent generative adversarial learning [48], as shown in Figure 3. In certain circumstances, the agents learned to exploit the physics of the environment to get on top of boxes and move around the environment "surfing" on a box, or even launching themselves into the air and successfully "seeking" the hiders mid-flight.

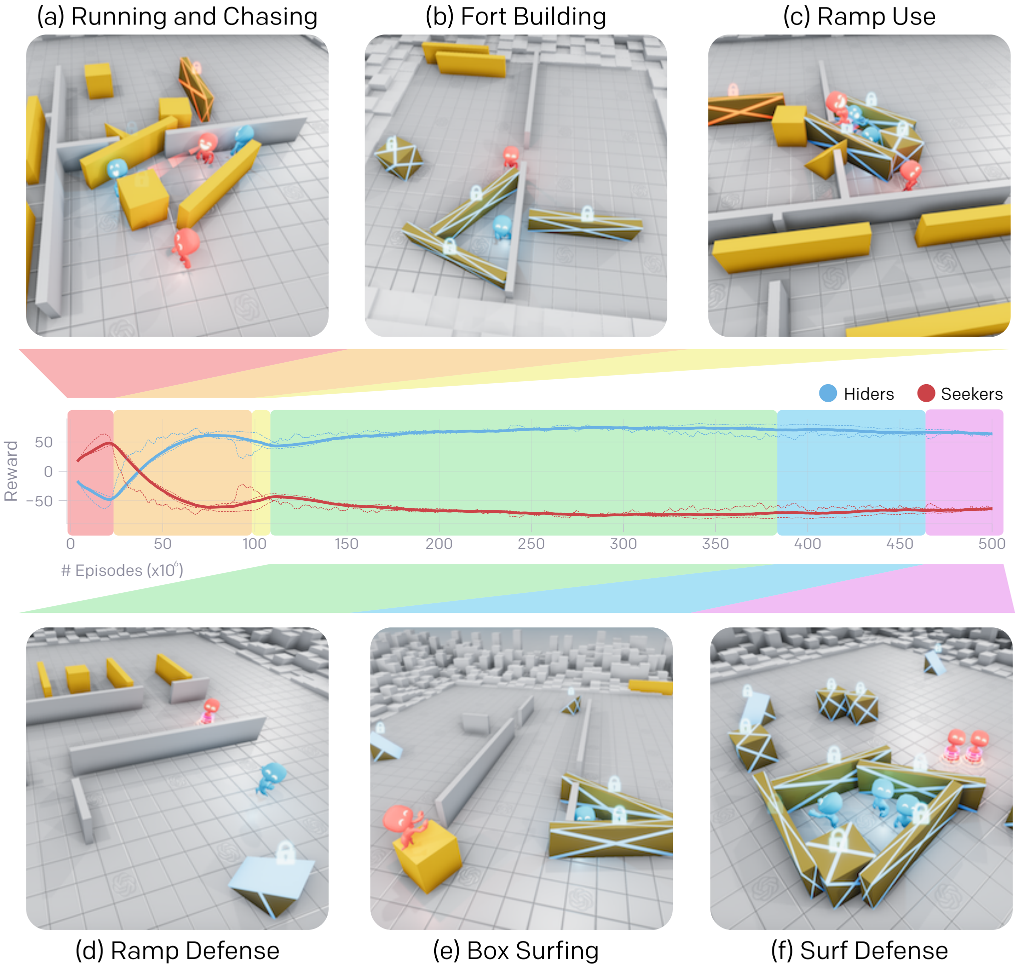

Figure 3. Visualisations of multi-agent adversarial learning and adaptations in strategy in OpenAI's hide-and-seek game [48]. Panels demonstrate the different strategies developed by the AI agents.

In another setting, it could be considered that the most impressive accomplishment of AlphaGo [49] is not that it can beat human grandmasters, but that the agent was able to develop never-before-seen play styles and strategies, even in a game invented millennia ago. As examples, this can be observed in the commentary from Go experts from the March 2016 challenge matches between Lee Sedol, one of the best players in the world, and AlphaGo, highlighted in the AlphaGo movie [50] (Match 1: 39:00–41:09; Match 2: 49:30–54:55; Match 4: 1:06:50–1:11:48 (AlphaGo resigns); Match 5: 1:17:02–1:23:17). In some of the cases, commentators first express confusion and/or consider the move to be an error, with the true implications only realised much later; e.g., in the Match 5 commentary: "Are we seeing another short circuit?", "There's no reason for white to be playing that move. It's a bad move. . . ", "We all say some of AlphaGo's moves are so weird and strange, and maybe mistakes. But after a game is finished, we have to doubt ourselves, our judgment", "I think it's important to study more about AlphaGo's mistake-like moves. Then maybe we can adjust our knowledge about Go", "It's playing some moves that are not really necessary". In other cases, the awe and surprise of the move is nearly immediate (e.g., Match 2: "Yeah, that's an exciting move. I think we are seeing an original move here. That is the kind of move that you play Go for."). AlphaGo can plan and consider many more moves ahead than any human, often 50–60 moves. AlphaGo even has the "awareness" that its move would be extremely unlikely to come from a human player (Match 2)—and similarly, had the awareness that Lee Sedol's critical move in Match 4 was also extremely unlikely. Since his matches against AlphaGo, Lee Sedol has retired from Go, shortly after the unveiling of AlphaGo Zero—which was able to beat the earlier AlphaGo (Lee) 100 games to zero [51]. Lee directly cited the AI agents as his reason for retiring: "With the debut of AI in Go games, I've realized that I'm not at the top even if I become the number one through frantic efforts. Even if I become the number one, there is an entity that cannot be defeated". [52] Where AlphaGo was trained on matches from expert players and followed by self-play, AlphaGo Zero began with only the game rules and did not have access to any human experiences. The report on AlphaGo Zero shows how, with continuing learning (i.e., hours of training), the agent is able to develop known Go joseki (patterns of moves) and in some cases mature past them and generate new, unrecorded winning joseki (see [51], Extended Data Figures 2 and 3). AlphaGo Zero has since been superseded by AlphaZero, where the AI agent has been generalised to be able to play chess and shogi at expert levels, as well as Go [53].

Like AlphaGo, AlphaStar has also been observed to develop novel strategies not yet explored by human players [8,17]. One of the expert StarCraft players reflected: "I was surprised by how strong the agent was, AlphaStar takes well-known strategies and turns them on their head. The agent demonstrated strategies I hadn't thought of before, which means there may still be new ways of playing the game that we haven't fully explored yet" [19]. In comparison to humans, AlphaStar appears more content to strategically kill a few units and immediately back away, rather than going "all in". More generally, AlphaStar is more decisive about when to engage in an attack, while the opposing humans have more difficulty deciding when to begin a conflict (e.g., back-and-forth movement showing hesitancy in committing to an attack). Watching replays since AlphaStar has had the camera constraint from AlphaStar's perspective are somewhat jarring as it often rapidly cycles between groups of units in different parts of the map, issuing a specific action command, then changing to another set of units within just a few seconds—e.g., no decision has to be made after the camera is changed; the action for those units has already been determined and simply was yet to be issued. More recent, individual games of AlphaStar also include odd behaviour that seems to be non-adaptive; though like AlphaGo, this may be preparing for situations we cannot identify. For instance, AlphaStar has been observed lifting and moving Terran buildings without an apparent goal and, in other instances, massing units that have limited utility (and are not sensible counters to the player). There are now more variants of AlphaStar playing against players on public servers, whereas the set of available play styles was initially more constrained and could only be played against in invitation-only demonstrations.

5. Conclusions

The discussion and advancement of AI technology is an on-going topic and ever evolving, with *What Computers Can't Do* [54] and *Rebooting AI* [55] serving as comprehensive overviews.

If the intent is to model the full agent capabilities, then AI agents must include, for example, computer vision and robotic actuators to appropriately see and act within the world. For instance, AlphaGo does not actually move the pieces on the board itself, though some are advancing the field in this direction, e.g., [39] is a one-handed robot that can solve the Rubik's Cube. If this is not the intended goal and only an aspect of agent behaviour is intended, then accommodations must also be included to constrain the behaviour to parameters that are meaningful and relevant to the environmental conditions. While this limitation hopefully is considered reasonable, this is not the current norm. Moreover, this approach will also help bring us closer to explainable AI—where humans can understand the inner workings of a "glass box" rather than the "black box" that underlies an AI agent's decision process [56–59]. For us to learn from AI developments, this is critical.

Some have posited a future where "robot psychologist" is a potential profession [60–63]. That is, a psychologist for robots, not a robot that acts as a psychologist; the goal here is to understand how robots think. The underlying "thinking" of object detection algorithms can now be somewhat explored using neural feature visualisation and inversion [64–66]. Before this hypothetical future can be realised, we need to make considerations about the tractable limitations in our AI; currently, a notable factor in the improvement of machine learning performance can be attributed to the use of increasing amounts of compute time and cores, not just more advanced agent architectures [67,68]. However, maybe, this future is not as distant as one may have previously thought.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
APM	Actions per minute
CFOP	Cross, F2L (first two layers), OLL (orient last layer), PLL (permute last layer); Rubik's Cube solving method
RTS	Real-time strategy
TAS	Tool-assisted speed-run

References

1. Whiteson, S.; Tanner, B.; White, A. Report on the 2008 Reinforcement Learning Competition. *AI Mag.* **2010**, *31*, 81–94. [CrossRef]
2. Togelius, J.; Karakovskiy, S.; Koutnik, J.; Schmidhuber, J. Super Mario evolution. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Games, Milano, Italy, 7–10 September 2009; pp. 156–161. [CrossRef]
3. Karakovskiy, S.; Togelius, J. The Mario AI Benchmark and Competitions. *IEEE Trans. Comput. Intell. AI Games* **2012**, *4*, 55–67. [CrossRef]
4. Bellemare, M.G.; Naddaf, Y.; Veness, J.; Bowling, M. The Arcade Learning Environment: An evaluation platform for general agents. *J. Artif. Intell. Res.* **2013**, *47*, 253–279. [CrossRef]
5. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540.
6. Rocki, K.M. Nintendo Learning Environment. 2019. Available online: https://github.com/krocki/gb (accessed on 9 May 2020). [CrossRef]
7. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
8. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354. [CrossRef] [PubMed]
9. Dann, M.; Zambetta, F.; Thangarajah, J. Deriving subgoals autonomously to accelerate learning in sparse reward domains. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 881–889. [CrossRef]
10. Ecoffet, A.; Huizinga, J.; Lehman, J.; Stanley, K.O.; Clune, J. Go-Explore: A new approach for hard-exploration problems. *arXiv* **2019**, arXiv:1901.10995.

11. Lewis, J.; Trinh, P.; Kirsh, D. A corpus analysis of strategy video game play in Starcraft: Brood War. In Proceedings of the Annual Meeting of the Cognitive Science Society, Boston, MA, USA, 20–23 July 2011; Volume 33, pp. 687–692.
12. Ontanon, S.; Synnaeve, G.; Uriarte, A.; Richoux, F.; Churchill, D.; Preuss, M. A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft. *IEEE Trans. Comput. Intell. AI Games* **2013**, *5*, 293–311. [CrossRef]
13. Robertson, G.; Watson, I. A review of real-time strategy game AI. *AI Mag.* **2014**, *35*, 75. [CrossRef]
14. Baumgarten, R. Infinite Mario AI. 2009. Available online: https://www.youtube.com/watch?v=0s3d1LfjWCI (accessed on 25 July 2019).
15. Jaderberg, M. AlphaStar Agent Visualisation. 2019. Available online: https://www.youtube.com/watch?v=HcZ48JDamyk (accessed on 25 July 2019).
16. Lee, T.B. An AI Crushed Two Human Pros at StarCraft—But It Wasn't a Fair Fight. *Ars Technica*. 2019. Available online: https://arstechnica.com/gaming/2019/01/an-ai-crushed-two-human-pros-at-starcraft-but-it-wasnt-a-fair-fight (accessed on 30 January 2019).
17. Heijnen, S. StarCraft 2: Lowko vs AlphaStar. 2019. Available online: https://www.youtube.com/watch?v=3HqwCrDBdTE (accessed on 22 November 2019).
18. Thompson, J.J.; Blair, M.R.; Henrey, A.J. Over the Hill at 24: Persistent Age-Related cognitive-motor decline in reaction times in an ecologically valid video game task begins in early adulthood. *PLoS ONE* **2014**, *9*, e94215. [CrossRef]
19. Vinyals, O.; Babuschkin, I.; Chung, J.; Mathieu, M.; Jaderberg, M.; Czarnecki, W.M.; Dudzik, A.; Huang, A.; Georgiev, P.; Powell, R.; et al. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. 2019. Available online: https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/ (accessed on 24 January 2019).
20. DeepMind. StarCraft II Demonstration. 2019. Available online: https://www.youtube.com/watch?v=cUTMhmVh1qs (accessed on 25 January 2019).
21. Zador, A.M. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nat. Commun.* **2019**, *10*, 3770. [CrossRef]
22. LeDoux, J. *The Deep History of Ourselves: The Four-Billion-Year Story of How We Got Conscious Brains*; Viking: New York, NY, USA, 2019.
23. Risi, S.; Preuss, M. Behind DeepMind's AlphaStar AI that reached grandmaster level in StarCraft II: Interview with Tom Schaul, Google DeepMind. *Kunstl. Intell.* **2020**, *34*, 85–86. [CrossRef]
24. High Score. Seaquest (Atari 2600 Expert/A) High Score: 276,510 Curtferrell (Camarillo, United States). Available online: http://highscore.com/games/Atari2600/Seaquest/578 (accessed on 29 May 2019).
25. TASVideos. [2599] A2600 Seaquest (USA) "Fastest 999999" by Morningpee in 01:39.8. Available online: http://tasvideos.org/2599M.html (accessed on 29 May 2019).
26. High Score. Kangaroo (Atari 2600) High Score: 55,600 BabofetH (Corregidora, Mexico). Available online: http://highscore.com/games/Atari2600/Kangaroo/652 (accessed on 29 May 2019).
27. Toromanoff, M.; Wirbel, E.; Moutarde, F. Is deep reinforcement learning really superhuman on Atari? Leveling the playing field. *arXiv* **2019**, arXiv:1908.04683.
28. Hessel, M.; Modayil, J.; Hasselt, V.H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, G.M.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 3215–3222.
29. Kapturowski, S.; Ostrovski, G.; Dabney, W.; Quan, J.; Munos, R. Recurrent experience replay in distributed reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
30. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *arXiv* **2020**, arXiv:1911.08265.
31. Korf, R.E. Sliding-tile puzzles and Rubik's Cube in AI research. *IEEE Intell. Syst.* **1999**, *14*, 8–12.
32. El-Sourani, N.; Hauke, S.; Borschbach, M. An evolutionary approach for solving the Rubik's Cube incorporating exact methods. *Lect. Notes Comput. Sci.* **2010**, *6024*, 80–89. [CrossRef]
33. Agostinelli, F.; McAleer, S.; Shmakov, A.; Baldi, P. Solving the Rubik's Cube with deep reinforcement learning and search. *Nat. Mach. Intell.* **2019**, *1*, 356–363. [CrossRef]

34. Rubik's Cube You Can Do the Rubik's Cube. 2020. Available online: https://www.youcandothecube.com/solve-the-cube/ (accessed on 5 April 2020).
35. Ruwix. Different Rubik's Cube Solving Methods. Available online: https://ruwix.com/the-rubiks-cube/different-rubiks-cube-solving-methods/ (accessed on 16 April 2020).
36. Thistlewaite, M.B. 45–52 Move Strategy for Solving the Rubik's Cube. Technical Report, University of Tennessee in Knoxville. 1981. Available online: https://www.jaapsch.net/puzzles/thistle.htm (accessed on 25 April 2020).
37. Rokicki, T.; Kociemba, H.; Davidson, M.; Dethridge, J. God's Number Is 20. 2010. Available online: http://cube20.org (accessed on 25 April 2020).
38. Yang, B.; Lancaster, P.E.; Srinivasa, S.S.; Smith, J.R. Benchmarking robot manipulation With the Rubik's Cube. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2094–2099. [CrossRef]
39. OpenAI; Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; et al. Solving Rubik's Cube with a Robot Hand. *arXiv* **2019**, arXiv:1910.07113.
40. TASVideos. Tool-Assisted Game Movies: When Human Skills Are Just Not Enough. Available online: http://tasvideos.org (accessed on 29 May 2019).
41. Madan, C.R. Augmented memory: A survey of the approaches to remembering more. *Front. Syst. Neurosci.* **2014**, *8*, 30. [CrossRef]
42. LiMieux, P. From NES-4021 to moSMB3.wmv: Speedrunning the serial interface. *Eludamos* **2014**, *8*, 7–31.
43. Potter, P. Saving Milliseconds and Wasting Hours: A Survey of Tool-Assisted Speedrunning. *Electromagn. Field* **2016**. Available online: https://www.youtube.com/watch?v=6uzWxLuXg7Y (accessed on 16 May 2019).
44. Chrabąszcz, P.; Loshchilov, I.; Hutter, F. Back to basics: Benchmarking canonical evolution strategies for playing Atari. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 13–19 July 2018; pp. 1419–1426. Available online: https://www.youtube.com/watch?v=meE5aaRJ0Zs (accessed on 2 March 2018). [CrossRef]
45. Sampson, G. Q*bert Scoring Glitch on Console. 2018. Available online: https://www.youtube.com/watch?v=VGyeUuysyqg (accessed on 22 July 2020).
46. Murphy, T. The first level of Super Mario Bros. is easy with lexicographic orderings and time travel…after that it gets a little tricky. In Proceedings of the 2013 SIGBOVIK Conference, Kaohsiung, Taiwan, 1 April 2013; pp. 112–133. Available online: http://tom7.org/mario/ (accessed on 22 July 2020).
47. Lehman, J.; Clune, J.; Misevic, D.; Adami, C.; Altenberg, L.; Beaulieu, J.; Bentley, P.J.; Bernard, S.; Beslon, G.; Bryson, D.M.; et al. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *arXiv* **2019**, arXiv:1803.03453.
48. Baker, B.; Kanitscheider, I.; Markov, T.; Wu, Y.; Powell, G.; McGrew, B.; Mordatch, I. Emergent tool use from multi-agent autocurricula. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26 April–1 May 2020. Available online: https://iclr.cc/virtual_2020/poster_SkxpxJBKwS.html (accessed on 5 May 2020).
49. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
50. DeepMind. AlphaGo: The Movie. 2017. Available online: https://www.youtube.com/watch?v=WXuK6gekU1Y (accessed on 14 August 2020).
51. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
52. Yonhap News Agency. Go Master Lee Says He Quits Unable to Win over AI Go Players. 2019. Available online: https://en.yna.co.kr/view/AEN20191127004800315 (accessed on 15 August 2020).
53. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **2018**, *362*, 1140–1144. [CrossRef]
54. Dreyfus, H.L. *What Computers Can't Do*; Harper & Row: New York, NY, USA, 1972.
55. Marcus, G.; Davis, E. *Rebooting AI*; Pantheon: Roman, Italy, 2019.

56. Miller, T.; Howe, P.; Sonenberg, L. Explainable AI: Beware of Inmates Running the Asylum. In Proceedings of the IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI), Melbourne, Australia, 20 August 2017; pp. 36–42. Available online: https://people.eng.unimelb.edu.au/tmiller/pubs/explanation-inmates.pdf (accessed on 12 May 2020).
57. Goebel, R.; Chander, A.; Holzinger, K.; Lecue, F.; Akata, Z.; Stumpf, S.; Kieseberg, P.; Holzinger, A. Explainable AI: The New 42? *Lect. Notes Comput. Sci.* **2018**, *11015*, 295–303. [CrossRef]
58. Holzinger, A. From Machine Learning to Explainable AI. In Proceedings of the IEEE 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), Kosice, Slovakia, 23–25 August 2018. [CrossRef]
59. Peters, D.; Robinson, K.V.D.; Calvo, R.A. Responsible AI–Two Frameworks for Ethical Design Practice. *IEEE Trans. Technol. Soc.* **2020**, *1*, 34–47. [CrossRef]
60. Asimov, I. *I, Robot*; Gnome Press: New York, NY, USA, 1950.
61. Gerrold, D. *When HARLIE Was One*; Ballantine Books: New York, NY, USA, 1972.
62. Čapek, K. *R.U.R.: Rossum's Universal Robots*; Project Gutenberg: Salt Lake City, UT, USA, 1921. Available online: http://www.gutenberg.org/files/59112/59112-h/59112-h.htm (accessed on 28 October 2019).
63. Gold, K. *Choice of Robots*; Choice of Games: San Franciscio, CA, USA, 2014. Available online: https://www.choiceofgames.com/robots/ (accessed on 31 August 2015).
64. Olah, C.; Mordvintsev, A.; Schubert, L. Feature Visualization. *Distill* **2017**. [CrossRef]
65. Olah, C.; Satyanarayan, A.; Johnson, I.; Carter, S.; Schubert, L.; Ye, K.; Mordvintsev, A. The Building Blocks of Interpretability. *Distill* **2018**. [CrossRef]
66. Carter, S.; Armstrong, Z.; Schubert, L.; Johnson, I.; Olah, C. Exploring Neural Networks with Activation Atlases. *Distill* **2019**. [CrossRef]
67. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
68. Xu, X.; Ding, Y.; Hu, S.X.; Niemier, M.; Cong, J.; Hu, Y.; Shi, Y. Scaling for edge inference of deep neural networks. *Nat. Electron.* **2018**, *1*, 216–222. [CrossRef]