AppleGameEnv: Mastering Massive Action Space in a Non-Tabular RL Environment

Sangoh Kim[*]
KAIST
tkddh1109@kaist.ac.kr

Sejong Kim[*]
KAIST
kingsj@kaist.ac.kr

Yuseop Lee [*]
KAIST
cats2969@kaist.ac.kr

Abstract

Reinforcement learning (RL) has proven successful in large state space grid games like Go, but environments with massive and dynamic action spaces present a significant challenge. Inspired by a popular online game, we introduce **AppleGameEnv**, a simple yet novel *non-tabular, grid-based* environment featuring a *large action space*. We benchmarked 8 standard RL algorithms, 2 non-learning baselines, human players, and a Language Generalized Policy Iteration (LGPI) agent that leverages a Large Language Model (LLM) for interpretable decision-making. We utilized action masking for efficient learning and evaluated all agents on a standardized set of 100 golden grids. Our results reveal a surprising gap in the capabilities of current RL methods. A simple rule-based heuristic significantly outperformed all trained RL agents in both mean score and average win rate. While DQN-based agents learned stable and effective strategies, policy-based and actor-critic methods struggled to surpass the random baseline. Furthermore, LGPI provided spatial reasoning for its decisions, highlighting a path toward explainable AI in complex domains. This presents **AppleGameEnv** as a challenging benchmark in which novel approaches are needed to bridge the gap in large action space problems, while acknowledging the tradeoff between optimality and interpretability.

1 Introduction

Reinforcement Learning (RL) has demonstrated effectiveness in addressing complex problems involving large state-space environments. Notably, numerous studies have successfully applied RL algorithms to solve grid-based games such as Chess Silver et al. [2017], Go Silver et al. [2016], and Atari games Mnih et al. [2013]. These environments, however, share a common limitation: their action spaces are relatively small and structured. Consequently, existing benchmarks do not adequately reflect the complexity and scale of real-world problems, where large, non-tabular action spaces frequently occur. Motivated by this gap, we introduce **AppleGameEnv**, a novel RL benchmark designed specifically to address large action spaces. In this study, we applied three primary categories of RL algorithms—*policy-based, value-based, and actor-critic*—to the environment. Our experiments demonstrated meaningful performance improvements across these methodologies. Additionally, we implemented a Natural Language Reinforcement Learning (NLRL) algorithm to enhance interpretability and reduce the black-box nature of traditional RL methods.

2 Related Work

The seminal work by Mnih et al. [2013] demonstrated significant effectiveness of reinforcement learning (RL) algorithms in Atari game environments. However, Atari games have a small state space (128) and action space (18), compared to AppleGame's vastly larger configuration space of

[*] All authors contributed equally.

Preprint. Under review.

10^{170}states and 170×170 actions, and thus fall short of capturing real-world complexity. Moreover, previous research primarily focused on demonstrating algorithmic effectiveness (Dulac-Arnold et al. [2016]), without thoroughly investigating the underlying strategies, mechanisms, or rationale for action selection within these algorithms. Consequently, insights into why specific strategies or actions were selected by the RL agents remained unexplored. Our work effectively explores these areas in a simple yet sophisticated grid game environment, following the works of Engelhardt et al. [2024].

3 Methods

Because no existing RL environment supported AppleGame, we implemented the game dynamics using Pygame and Gymnasium. To improve agent performance, we then incorporated a valid action masking mechanism. Finally, to enhance interpretability, we applied Language Generalized Policy Iteration (LGPI; Feng et al. [2025]).

3.1 Benchmark Environment: AppleGameEnv

AppleGame (Figure 3) is a popular puzzle game on the App Store and Google Play. The player selects a rectangular region of the grid such that the sum of all apples in that region equals 10. When the region is cleared, the score increases with the number of cells removed. The objective is to clear as many apples as possible within a 120-second time limit.

We developed a new RL environment, `AppleGameEnv`, on top of PyGame and Gymnasium. This environment supports both human and agent play, and provides action masking. Following the 10×17 grid of the original game, we formalize the state and action spaces:

$$S = \{(x, y, n) : 0 \leq x \leq 9, 0 \leq y \leq 16, 0 \leq n \leq 9\}$$

$$A = \{(x, y, w, h) : x \geq 0, y \geq 0, 0 < x + w \leq 9, 0 < y + h \leq 16\}$$

As shown, the action space is extremely large, causing naive RL agents to suffer severe performance degradation. Our demonstration video and slides are available at here.

3.2 Action Masking

Initially, we applied standard RL agents directly, but they frequently chose invalid actions (regions summing to a value other than 10) and achieved only 3–4 points on average over the full time limit. To address this, we integrated an action-masking strategy (Huang and Ontañón [2022]).

As illustrated in Figure 1, the environment automatically masks out invalid actions before the agent computes its final probability distribution. Since valid actions depend on the current grid state, action masking is a well-suited technique for dynamically changing action sets in AppleGame.

3.3 Language Generalized Policy Iteration

Language Generalized Policy Iteration (LGPI, Feng et al. [2025]) enhances standard policy iteration by integrating natural-language reasoning at each decision step, while conforming to computational constraints. As shown in Figure 4, LGPI follows 3 steps:

1. For each feasible action, the agent formulates a prompt containing the current state, the action's immediate outcome, and queries the LLM in a zero-shot fashion. The model returns a explanation—detailing why the action may be beneficial/risky—and a concise natural-language value summary.
2. The agent collects all narrative assessments for a given action and merge them into a single explanation with LLM. This step highlights the most salient pros and cons observed across the repeated evaluations.
3. Once each action has a unified explanation, the agent presents all such explanations to the LLM simultaneously. The model then compares each action's benefits and drawbacks and selects the best action index. The output includes both the chosen action and a natural-language justification, making the policy update fully interpretable.

By leveraging an off-the-shelf LLM without additional training, LGPI maintains feasibility under limited computational budgets. Because every decision is accompanied by a text-based rationale, researchers can readily audit and refine the agent's reasoning.

4 Experiments

4.1 Experimental Setup

We conducted experiments on the proposed **AppleGameEnv**. Training and testing sessions were standardized to 1,000 episodes for each RL algorithm, which corresponds to approximately 30,000 time steps due to the variable action space (1 to 60 actions per state). For function approximation, hyperparameters (e.g. learning rate, epsilon decay, experience replay buffer size, discount factor γ) adhered closely to their original implementations, with the neural network architecture fixed due to computational constraints.

Experiments were performed in a notebook-based environment on Google Colab's free tier, utilizing NVIDIA T4 GPUs. A fully reproducible pipeline, including library versions and model checkpoints, is available at `https://github.com/thomaskim1130/AppleGameRL`.

4.2 Implementation Details

Evaluation Metrics Following conventions in video game RL research, we report **mean scores** and **win rates** across agents. Win rates, excluding draws, provide insight into performance in the stochastic and expansive AppleGameEnv, where clearing all apples constitutes a win.

Golden Grids We generated a fixed set of 100 random 10×17 grids, termed **golden grids**, for consistent evaluation across all agents. Due to resource constraints, particularly for neural networks and language models in LGPI, all reported results are based on performance over these grids.

RL Agents We benchmarked 8 RL agents across 3 categories: *value-based* (GPI Sutton and Barto [1998], DQN Mnih et al. [2015], DuelingDQN Wang et al. [2016]), *policy-based* (REINFORCE Williams [1992], DDPG Lillicrap et al. [2016]), and *actor-critic* (QAC Konda and Tsitsiklis [2003], A2C Mnih et al. [2016], PPO Schulman et al. [2017]). Each agent's implementation followed its original pseudocode, with neural network architectures tailored for AppleGameEnv.

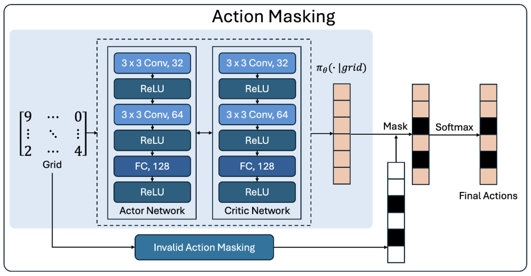


Figure 1: Architecture of the actor-critic networks with action masking.

As shown in Figure 1, the input to the function approximator is a normalized 10×17×1 grid state. The Actor (policy) network's first hidden layer convolves 32 3×3 filters with stride 1 and padding 1, followed by a ReLU nonlinearity. The second hidden layer convolves 64 3×3 filters with stride 1 and padding 1, again followed by ReLU. The output is flattened and passed through a fully connected layer with 128 ReLU units, followed by a linear layer outputting logits for up to 1,000 valid actions, with action masking and softmax to handle variable action spaces. In testing, actions are selected deterministically via argmax. The Value (critic) network mirrors this architecture but outputs a single state value. Both networks use Xavier uniform weight initialization and zero-initialized biases.

GreedyAgent and HeuristicAgent To establish baselines, we implemented two non-learning agents: *GreedyAgent* and *HeuristicAgent*.

GreedyAgent selects valid actions uniformly at random after applying the action mask, serving as a minimal benchmark to assess whether RL agents learn meaningful strategies. HeuristicAgent employs a two-stage heuristic inspired by online communities: (1) early game, prioritize clearing smaller groups of adjacent apples (e.g., 9-1, 8-2); (2) endgame, maximize apples cleared per move (e.g., 3-3-4, 1-2-3-4). This agent represents an optimal human-like strategy without training.

LGPIAgent We followed the original structure in NLRL and adapted the prompts for AppleGameEnv (Figure 4). Setting the system prompt to explain AppleGameEnv, the state of the grid is inputted. For action evaluation, 1-step TD was used instead of n-step TD from NLRL to match the dynamic action space with up to 50 valid actions. For all single valid actions, the thoughts or 'language value functions' were aggregated line-by-line. Then, we applied listwise ranking to choose the next action. Actual outputs are shown in Figure 5.

The exact prompts can be found on our GitHub. Due to the complexity and cost of each episode, the model used is *gpt-4.1-nano-2025-04-14* with temperature set to 0.7. Further analyses on cost and time taken are described in Table 1.

Human and HumanExpert To contextualize RL performance against human players, we utilized AppleGameEnv's "human play" mode with a 120-second time limit per grid, declaring "game over" when time expires or all apples are cleared. Three researchers (n=3) with varying expertise—*Novice* (<1 month experience), *Casual* (1 year), and *Expert* (3+ years)—played non-overlapping subsets of the 100 golden grids (33 grids each). The combined results form the *Human* agent, while grids 67–100, played by the Expert, constitute the *HumanExpert* agent. The win rate of HumanExpert over Human is set to 100% for reference.

5 Results

We report performance metrics of 8 RL agents(*GPI, DQN, DuelingDQN, REINFORCE, DDPG, QAC, A2C, PPO*), 2 non-learning agents(*Greedy, Heuristic*), and 2 human agents(*Human, HumanExpert*). We additionally provide empirical analysis on the interpretability of agents, especially on their strategies for solving AppleGameEnv.

5.1 Quantitative Analysis

After training each agent accordingly, we tested each model through golden grids 1 to 100. The mean scores and standard deviations are shown in Figure 2a. Since the golden grids are fixed, we compared all the agents on each grid to obtain the win rates in Figure 2b.

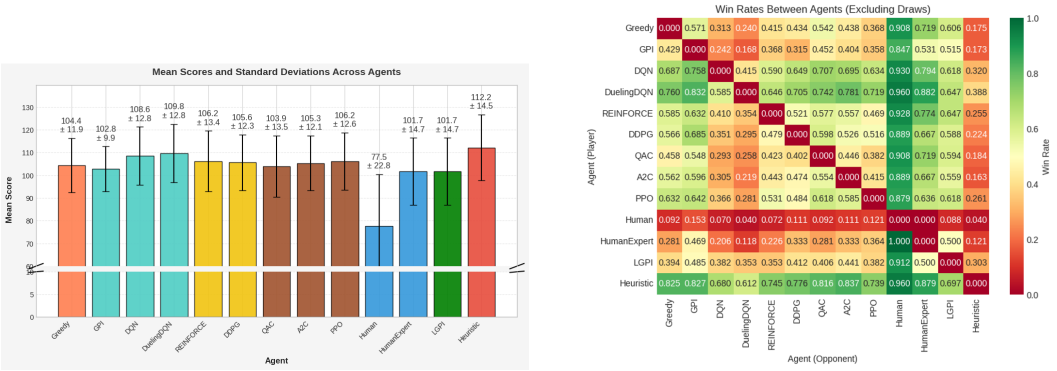
In mean scores, HeuristicAgent performed exceedingly well, scoring 112.2. DuelingDQNAgent and DQNAgent fell slightly behind, scoring 109.8 and 108.6, respectively. Other baseline agents showed a mediocre score, while GPIAgent, LGPIAgent, and human agents lagged behind even the random GreedyAgent's score of 104.4. (Note the maximum score obtainable is 170)

All agents were successful at defeating human agents, displaying the effectiveness of action masking and RL algorithms. As demonstrated by Mnih et al. [2015], DQN-based agents demonstrated stable convergence and high performance, thanks to its replay buffer. However, policy-based agents were unstable during the learning phase and did not make a huge improvement over the GreedyAgent. The most popular algorithm PPO underperformed, mainly due to the sparse nature of AppleGameEnv. Interestingly, GPI and LGPI worsened during learning, converging towards a suboptimality. This can be attributed to the large nature of AppleGameEnv, showing limitations of traditional value-based methods.

However, a simply-implemented HeuristicAgent came on top with a **85.1%** win rate on average. This hints at a *potential gap in research for large, dynamic action-space environments* like AppleGameEnv.

5.2 Qualitative Analysis

In the playing video(3.1), we can see real-time clips of the QACAgent without action masking, the DuelingDQNAgent solving grids 1 to 4, and the human playing version of AppleGameEnv.



(a) Mean scores and standard deviations of all 11 agents and 2 human agents.

(b) Win rates of agents vs. each other. Y-axis is the player, x-axis is the opponent. Diagonals are 0.0.

Figure 2: Performance of agents

Before applying action masking, the agent chose mostly invalid actions. This is because only about 50 actions (at most) out of 170*170 possible actions sum up to 10. While analysis of whether the agents can learn which actions are valid remains an interesting topic, action masking empirically showed significantly improved performance with efficient runtimes. As mathematically feasible as it is (Huang and Ontañón [2022]), applying action masking proved to be a viable, if not indispensable option for tackling a large action-space.

After learning, the DuelingDQNAgent as well as the DDPGAgent and PPOAgent, portrayed a clear strategy—clearing apples from left to right (or right to left), only contacting neighboring ones. This is a well-known tactic for human players of AppleGame, in order to create open space for more valid actions. Also, they showed knowledge of the sequence of actions, clearing apples with consideration of the next grid state. Such results further prove the effectiveness of RL in solving complex environments.

However, such interpretations of these neural network-based agents are solely based on human observation of their choices. On the contrary, LGPIAgent provided an in-depth reasoning for each action, as shown in Figure 5. It effectively evaluated each valid action acknowledging their horizontal/vertical traits, then compared them according to their potential of clearing more apples. Even without training, a pre-trained model like a LLM with prompt engineering displayed a distinct edge in terms of interpretability. The time and cost taken to run LGPIAgent are detailed in Table 1.

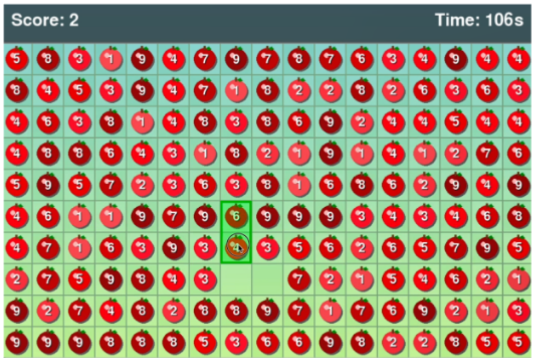
6 Conclusion

We introduced **AppleGameEnv**, a non-tabular, grid-based RL benchmark with a massive, dynamic action space. By employing *action masking*, agents were restricted to valid moves and greatly outperformed naive baselines—and even human players—yet our simple *HeuristicAgent* still surpassed all learned policies, exposing a key limitation of current RL methods. We also demonstrated *Language Generalized Policy Iteration* (LGPI) for transparent, human-readable decision rationales, trading some performance for interpretability.

Future work will focus on (1) designing scalable algorithms for large action spaces, (2) introducing hierarchical or more efficient value approximation strategies, and (3) blending LGPI's explainability with policy learning to balance optimality and transparency.

Contribution *Sejong Kim* set up AppleGameEnv to be compatible with OpenAI Gymnasium, then trained actor-critic agents. He performed extensive quantitative analyses on agents, and acted as the HumanExpert. He mainly set up this LaTeX document. *Sangoh Kim* implemented DQN and DuelingDQN algorithms, and adapted NLRL for LGPI. He performed qualitative analysis on their interpretability, created figures, and acted as the Human(Novice). *Yuseop Lee* trained policy-based agents, and conducted literature review on related works. He also created the presentation material, and acted as the Human(Casual).

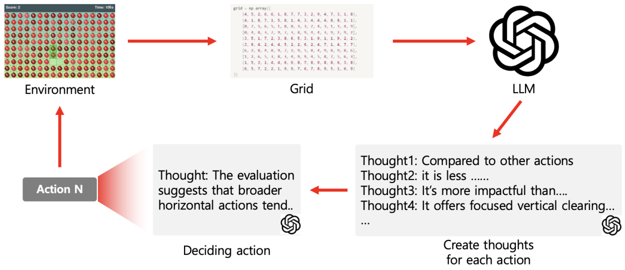
7 Supplementary Material



Score: 2 Time: 106s

5	8	3	1	9	4	7	9	7	8	7	6	3	4	9	4	4
8	4	5	3	9	4	7	1	8	2	2	8	2	6	3	6	3
4	6	3	8	1	4	8	3	8	6	9	2	4	4	5	4	4
4	8	8	6	4	3	1	8	2	1	9	1	4	1	2	7	6
5	9	5	7	2	3	6	3	8	1	6	8	6	2	9	4	9
4	6	1	1	9	7	9	6	9	9	9	3	4	3	4	6	8
4	7	1	6	3	9	3	4	3	5	6	2	6	5	7	9	5
2	7	5	9	8	4	3			7	2	1	5	4	6	2	1
9	2	7	4	8	2	8	8	9	7	1	7	6	9	2	1	3
9	9	9	8	8	8	5	3	6	6	9	8	2	2	8	5	5

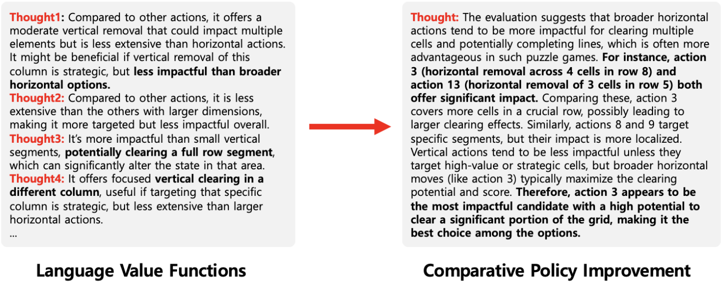
Figure 3: User interface of AppleGame. The green border indicates that the action 6+4 is valid.



Thought1: Compared to other actions
Thought2: it is less ......
Thought3: It's more impactful than....
Thought4: It offers focused vertical clearing...
...

Thought: The evaluation suggests that broader horizontal actions tend..

Figure 4: Pipeline for LGPIAgent.



Thought1: Compared to other actions, it offers a moderate vertical removal that could impact multiple elements but is less extensive than horizontal actions. It might be beneficial if vertical removal of this column is strategic, but **less impactful than broader horizontal options.**
Thought2: Compared to other actions, it is less extensive than the others with larger dimensions, making it more targeted but less impactful overall.
Thought3: It's more impactful than small vertical segments, **potentially clearing a full row segment**, which can significantly alter the state in that area.
Thought4: It offers focused **vertical clearing in a different column**, useful if targeting that specific column is strategic, but less extensive than larger horizontal actions.
...

Thought: The evaluation suggests that broader horizontal actions tend to be more impactful for clearing multiple cells and potentially completing lines, which is often more advantageous in such puzzle games. **For instance, action 3 (horizontal removal across 4 cells in row 8) and action 13 (horizontal removal of 3 cells in row 5) both offer significant impact.** Comparing these, action 3 covers more cells in a crucial row, possibly leading to larger clearing effects. Similarly, actions 8 and 9 target specific segments, but their impact is more localized. Vertical actions tend to be less impactful unless they target high-value or strategic cells, but broader horizontal moves (like action 3) typically maximize the clearing potential and score. **Therefore, action 3 appears to be the most impactful candidate with a high potential to clear a significant portion of the grid, making it the best choice among the options.**

Figure 5: Output (Chain-of-Thought) of LGPIAgent on golden grid 1.

Metric	Cost		Time	
	(USD)	(KRW)	(seconds)	(hours)
Total	1.27	1773.66	54 022.08	15.01
Average per grid	0.04	54.29	1637.03	0.45

Table 1: Summary of computational cost and time for LGPIAgent on 33 golden grids.

References

Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces, 2016. URL `https://arxiv.org/abs/1512.07679`.

Raphael Engelhardt, Ralitsa Raycheva, Moritz Lange, Laurenz Wiskott, and Wolfgang Konen. Ökolopoly: Case study on large action spaces in reinforcement learning. pages 109–123, 02 2024. ISBN 978-3-031-53965-7. doi: 10.1007/978-3-031-53966-4_9.

Xidong Feng, Bo Liu, Yan Song, Haotian Fu, Ziyu Wan, Girish A. Koushik, Zhiyuan Hu, Mengyue Yang, Ying Wen, and Jun Wang. Natural language reinforcement learning, 2025.

Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. In *The International FLAIRS Conference Proceedings*, volume 35, 2022.

Vijay R. Konda and John N. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1166, 2003.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016*, pages 1–14, 2016.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL `https://arxiv.org/abs/1312.5602`.

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. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 1928–1937. JMLR.org, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi: 10.1038/nature16961.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. 12 2017. doi: 10.48550/arXiv.1712.01815.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 1995–2003. JMLR.org, 2016.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.