International Journal of Internet, Broadcasting and Communication Vol.16 No.3 192-198 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.3.192

IJIBC 24-3-19

Multi-Agent Deep Reinforcement Learning for Fighting Game: A Comparative Study of PPO and A2C

Yoshua Kaleb Purwanto[1], Dae-Ki Kang[2,*],

[1] *Master Student, Department of Computer Engineering, Dongseo University, Busan, Korea*
[2] *Professor, Department of Computer Engineering, Dongseo University, Busan, Korea*
yoshuakaleb049@gmail.com, dkkang@dongseo.ac.kr

Abstract

This paper investigates the application of multi-agent deep reinforcement learning in the fighting game Samurai Shodown using Proximal Policy Optimization (PPO) and Advantage Actor-Critic (A2C) algorithms. Initially, agents are trained separately for 200,000 timesteps using Convolutional Neural Network (CNN) and Multi-Layer Perceptron (MLP) with LSTM networks. PPO demonstrates superior performance early on with stable policy updates, while A2C shows better adaptation and higher rewards over extended training periods, culminating in A2C outperforming PPO after 1,000,000 timesteps. These findings highlight PPO's effectiveness for short-term training and A2C's advantages in long-term learning scenarios, emphasizing the importance of algorithm selection based on training duration and task complexity. The code can be found in this link https://github.com/Lexer04/Samurai-Shodown-with-Reinforcement-Learning-PPO.

Keywords: *Multi-Agent Reinforcement Learning; Proximal Policy Optimization; Advantage Actor-Critic; Performance Evaluation*

1. Introduction

In recent years, Multi-Agent Deep Reinforcement Learning (MARL) has seen widespread application across various sectors including industrial processes and financial trading. MARL enables agents to learn and collaborate, enhancing task efficiency and decision-making. It simulates teamwork dynamics crucial for achieving cooperative objectives, as seen in scenarios like Hide-and-Seek where strategic coordination among agents is essential for success [2]. Additionally, MARL can be applied to competitive environments, evaluating algorithms' performance under adversarial conditions such as combat scenarios where agents compete against each other, analyzing their unique strategies and adaptability, particularly in managing resources like health points.

Video games serve as valuable environments for testing AI capabilities due to their dynamic and complex nature [1]. Techniques such as visual input to neural networks allow agents to perceive and respond to game

Manuscript Received: June. 7, 2024 / Revised: June. 14, 2024 / Accepted: June. 20, 2024
Corresponding Author dkkang@dongseo.ac.kr
Tel: +82-51-320-1724, Fax: +82-51-327-8955
Professor, Department of Computer Engineering, Dongseo University, Busan, Korea

states effectively [3]. The introduction of Arcade Learning Environment (ALE) has expanded these capabilities, providing a standardized platform for evaluating AI behavior across various games [4]. Games like Samurai Shodown present specific challenges for reinforcement learning due to their strategic gameplay, motivating research aimed at developing RL agents capable of mastering such complexities. This study employs Proximal Policy Optimization (PPO) and Advantage Actor-Critic (A2C) algorithms to train agents in Samurai Shodown, focusing on their performance, interaction with the environment, and adaptability over extended training periods. These experiments aim to provide insights into the strengths and weaknesses of PPO and A2C in navigating intricate gaming scenarios, highlighting their potential applications and optimizations in diverse real-world settings.

2. Related Work

2.1 Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning revolutionizes traditional reinforcement learning by involving multiple agents that interact within a shared environment. Each agent optimizes its policy based on the state of the environment and the actions of other agents, creating a dynamic system where cooperation, competition, and communication are pivotal [7]. MARL has garnered attention for its ability to address complex real-world challenges requiring decentralized decision-making and coordination among multiple entities [8].

The practical applications of MARL span diverse domains such as industrial automation, autonomous driving, and financial trading. In industrial automation, MARL coordinates multiple robotic arms on assembly lines to boost productivity and efficiency [9]. In autonomous driving, it enables vehicles to interact in real-time, enhancing traffic flow and safety [10]. In financial trading, MARL algorithms manage portfolios and react to market changes more adeptly compared to single-agent approaches.

Research in MARL has yielded various algorithms tailored for multi-agent systems, including cooperative and competitive learning methods [11]. These methods facilitate collaboration towards common goals or competition for individual rewards, each posing unique challenges such as non-stationarity due to changing dynamics and the intricate exploration-exploitation trade-off [12]. Innovations like centralized training with decentralized execution and communication protocols help mitigate these challenges, enhancing the stability and efficiency of MARL systems.

2.2 Proximal Policy Optimization

Policy gradient methods in reinforcement learning optimize the policy directly to maximize expected rewards by adjusting policy parameters based on action outcomes [13]. This approach is well-suited for complex environments with high-dimensional action spaces and stochastic policies. Trust region methods, on the other hand, maintain policy stability by limiting the extent of policy updates within a defined "trust region" [14]. This ensures that policy changes are controlled to prevent drastic performance degradation while allowing for gradual improvements in learning.

Proximal Policy Optimization enhances policy gradient methods by incorporating trust region constraints through a clipping mechanism [5]. The PPO clip algorithm modifies the objective function to penalize large policy updates, thereby ensuring that the new policy remains close to the old one. This strategy stabilizes learning and balances exploration of new strategies with exploitation of learned policies. PPO's approach mitigates the instability issues of traditional policy gradient methods, making it effective in achieving steady progress without the risk of performance collapse. Its simplicity and robustness have led to widespread adoption across various reinforcement learning applications, showcasing its versatility and effectiveness in diverse environments.

$$L^{CLIP}{}_{(\theta)} = \widehat{E_t}\left[\min\left(r_t(\theta)\widehat{A_t}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A_t}\right)\right] \quad (1)$$

2.3 Advantage Actor-Critic

Advantage Actor-Critic (A2C) is an influential algorithm in reinforcement learning that combines elements of both policy-based and value-based methods. As part of the Actor-Critic family, A2C assigns the actor to select actions based on a policy, while the critic evaluates these actions using a value function. A key feature of A2C is its use of the advantage function, which reduces variance in policy gradient updates by subtracting a baseline value (often the state value) from the action value [6]. This mechanism stabilizes learning by measuring the efficacy of actions relative to expected values in a given state, enabling more precise policy updates that enhance learning efficiency and performance [15].

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} E_{s,a\sim\pi_\theta}\left[\log \pi_\theta\,(a \mid s)A^{\pi}(s,a)\right] \quad (2)$$

A2C draws parallels with previous reinforcement learning architectures like Gorila (General Reinforcement Learning Architecture), which introduced distributed training across multiple workers to accelerate complex model learning [16]. By leveraging parallelized actor and critic learners, A2C optimizes training speed and stability, particularly beneficial in environments with high-dimensional state and action spaces. This capability has empowered A2C and its variants, such as A3C, to excel across diverse tasks including game playing, robotic control, and simulated environments. Their demonstrated ability to train agents to perform competitively or even surpass human-level performance underscores their effectiveness in dynamic and interactive settings [6].

3. Experiment Setup and Methodology

3.1 Setup Environment

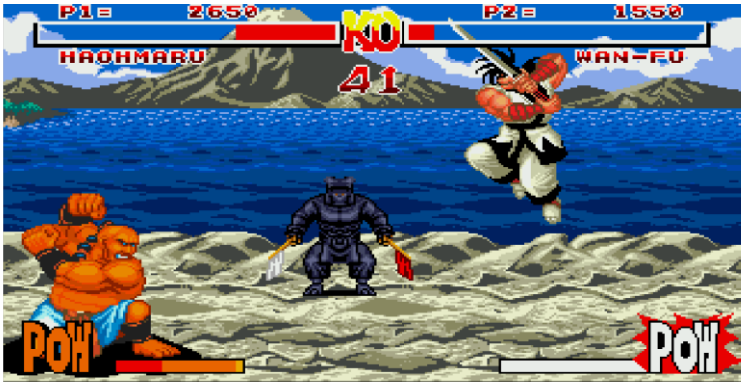


Figure 1. Player 1's Agent versus Player 2's Agent

The "Samurai Shodown" environment, integrated using the Gym Retro library for reinforcement learning, serves as a rich testbed due to its complex and dynamic nature in classic video games. This setup challenges RL agents with tasks like real-time decision-making, strategic planning, and opponent modeling, essential for training algorithms to optimize performance across various real-world applications.

In this environment, state representation crucially combines visual and RAM-based observations. Visual data, captured as image frames, offers insights into game state elements such as character positions, animation movements, and environmental details like game borders. Complementing this, RAM-based observations provide specific high-level information such as player and opponent health points (HP), derived from identified memory addresses within the game. This dual representation equips RL agents with detailed visual cues and numerical data, enhancing their ability to interpret and respond accurately to dynamic game scenarios.

The action space in "Samurai Shodown" is defined by a wide array of moves and combinations available to game characters. Utilizing the Gym Retro framework and Stable Baselines 3, the action space accommodates both discrete actions, such as single button presses or simple combinations, and multi-discrete actions that allow simultaneous pressing of multiple buttons [17][18]. This flexibility enables agents to execute diverse actions ranging from basic maneuvers to complex combo sequences, mirroring the full spectrum of gameplay mechanics present in the original Sega Genesis controller setup. The reward structure complements these capabilities by incentivizing effective gameplay strategies through feedback based on agent actions and performance. Rewards are tailored to specific in-game events and outcomes, such as successful attacks or significant game progress milestones, fostering the development of nuanced offensive and defensive tactics in RL agents.

3.2 Independent Learning

In this study, we designed separate training environments for the PPO (Proximal Policy Optimization) and A2C (Advantage Actor-Critic) algorithms, with agents (Player 1 and Player 2) competing against a computer-controlled enemy. The training duration is set to 200,000 timesteps, allowing sufficient exposure for the agents to learn the game's complexity and develop effective strategies. We selected Haohmaru and Wan-Fu from Samurai Shodown for their distinct combat styles—Haohmaru's speed and agility versus Wan-Fu's slower but powerful attacks. This setup helps assess the algorithms' adaptability to different fighting styles. After independent training, the PPO-trained and A2C-trained agents will compete against each other in a head-to-head battle to provide insights into their strengths and weaknesses in real-time combat. Observing their interactions and strategies will help determine which algorithm is more effective in adapting to opponents' tactics, managing resources, and executing complex maneuvers under pressure, highlighting the practical implications of using PPO versus A2C in similar environments.

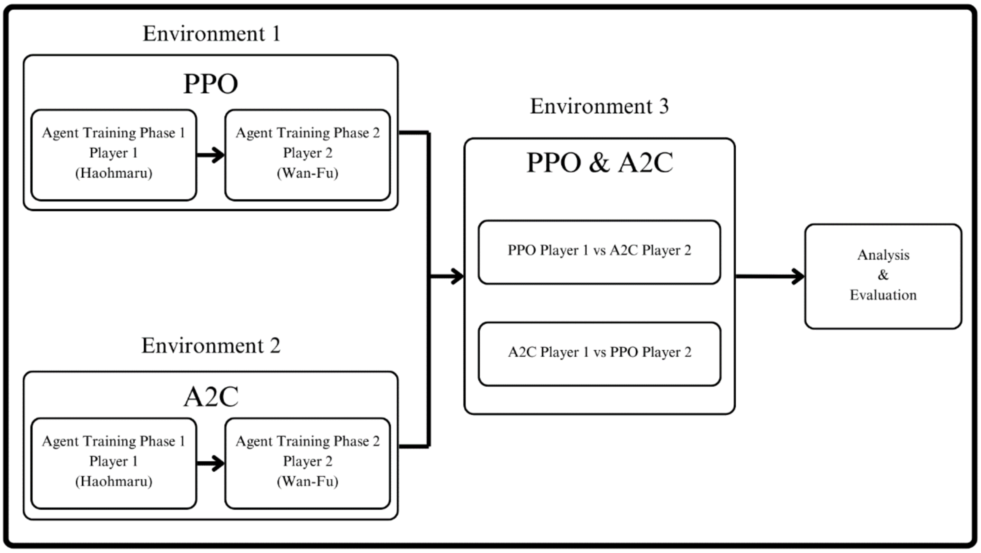
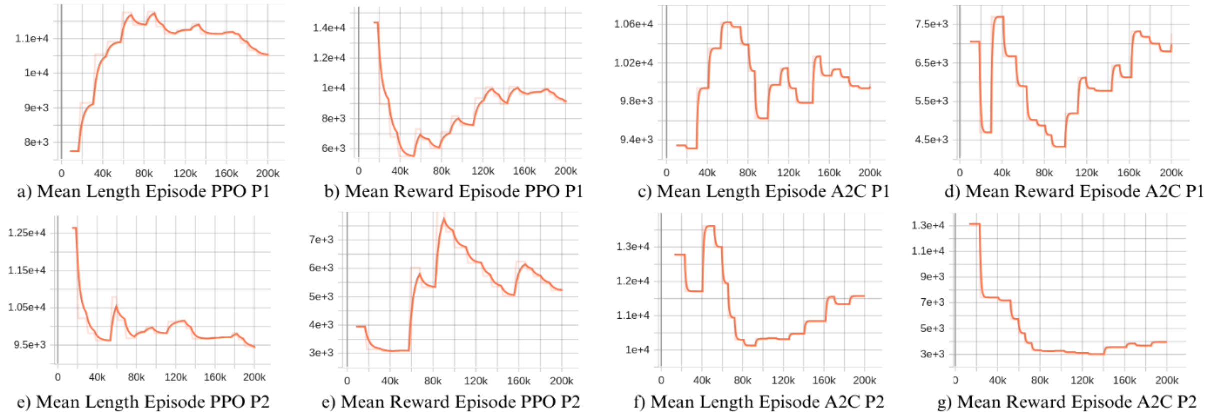


Figure 2. Block Diagram

4. Result and Discussion

Result. Based on reward accumulation over 200,000 timesteps, PPO Player 1 and Player 2 achieved average accumulated rewards of 9510 and 5357 per episode, respectively, while A2C Player 1 and Player 2 achieved 6894 and 3893. This disparity reflects PPO's conservative policy gradient method, ensuring gradual updates within a trust region for balanced exploration and exploitation. PPO's stable approach leads to consistent improvements in reward accumulation. In contrast, A2C uses advantage estimates for immediate policy updates based on feedback, resulting in more variable performance due to occasional exploration of suboptimal strategies, affecting its overall reward accumulation.



a) Mean Length Episode PPO P1 b) Mean Reward Episode PPO P1 c) Mean Length Episode A2C P1 d) Mean Reward Episode A2C P1

e) Mean Length Episode PPO P2 e) Mean Reward Episode PPO P2 f) Mean Length Episode A2C P2 g) Mean Reward Episode A2C P2

Figure 3. PPO and A2C 200.000 Timestep Mean Length and Reward

Table 1. Algorithm Fighting Results

Algorithm	Timesteps	P1 Win	P2 Win	Total Play
PPO vs A2C	200000	8	2	10
A2C vs PPO	200000	7	3	10
A2C vs PPO	1000000	10	0	10

Agent vs. Agent Competitive Evaluation. In the evaluation of PPO and A2C algorithms using Samurai Shodown characters Haohmaru and Wan-Fu, distinct learning approaches emerge. At 200,000 timesteps, PPO Player 1 outperformed A2C Player 1 with an 8-2 score, favoring PPO's fit for Haohmaru's fast-paced combat style. Conversely, A2C Player 1 won 7-3 against PPO Player 2, showcasing A2C's effectiveness with Wan-Fu's slower, high-damage strategy. By 1,000,000 timesteps, A2C Player 1 consistently outperformed PPO Player 2 with a 10-0 score, highlighting A2C's superior adaptation to Wan-Fu's gameplay. PPO struggled with Wan-Fu, likely due to its cautious policy updates hindering adaptability, emphasizing PPO's stability and A2C's responsiveness. The analysis reveals insights into PPO's initial success with Haohmaru but challenges with Wan-Fu's complexity over extended training. A2C, adept at both characters, demonstrated continuous improvement and outperformed PPO in prolonged scenarios. A2C's agile feedback-driven updates enabled better handling of variability. Future advancements could blend PPO's stability with A2C's adaptability for a more robust algorithm. Further exploration with diverse characters and game dynamics will enhance understanding and application of these algorithms in multi-agent deep reinforcement learning.

5. Conclusion

This research compares PPO and A2C algorithms in training characters for a fighting game. Initially, up to 200,000 timesteps, PPO achieved higher accumulated rewards per episode than A2C. PPO's stable updates within a trust region ensured steady learning and a balanced exploration-exploitation trade-off. However, beyond 500,000 timesteps, PPO's performance declined, revealing limitations in prolonged training. In contrast, A2C consistently improved, with A2C Player 1 maintaining higher rewards over time. A2C's advantage-based learning enabled adaptive decision-making, leading to A2C Player 1's decisive 10-0 victory over PPO Player 2 after 1,000,000 timesteps. While PPO excels in short-term stability, A2C's dynamic learning suits longer-term scenarios, highlighting the importance of algorithm choice based on training duration and task complexity.

Acknowledgement

This research was supported by “Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2023RIS-007) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2022R1A2C2012243).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," Nature, Feb. 2015.
DOI: https://doi.org/10.1038/nature14236

[2] B. Baker, I. Kanitschedier, T. Markov, et al., "Emergent Tool Use From Multi-Agent Autocurricula," Sep. 2019.
DOI: https://doi.org/10.48550/arXiv.1909.07528

[3] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Playing Atari with Deep Reinforcement Learning," Dec. 2013.
DOI: https://doi.org/10.48550/arXiv.1312.5602

[4] M. G. Bellemare, Y. Naddaf, J. Veness, et al., "The Arcade Learning Environment: An Evaluation Platform for General Agents," 2012.
DOI: https://doi.org/10.48550/arXiv.1207.4708

[5] J. Schulman, F. Wolski, P. Dhariwal, et al., "Proximal Policy Optimization Algorithms," Jul. 2017.
DOI: https://doi.org/10.48550/arXiv.1707.06347

[6] V. Mnih, A. P. Badia, M. Mirza, et al., "Asynchronous Methods for Deep Reinforcement Learning," 2016.
DOI: https://doi.org/10.48550/arXiv.1602.01783

[7] L. Busoniu, R. Babuska, B. De Schutter, "A comprehensive survey of multi agent reinforcement learning," 2008.
DOI: 10.1109/TSMCC.2007.913919

[8] P. Hernandez, B. Kartal, M. Taylor, "A survey and critique of multi agent deep reinforcement learning," Oct. 2018.
DOI: https://doi.org/10.1007/s10458-019-09421-1

[9] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," Oct. 1997.
DOI: https://dl.acm.org/doi/10.5555/284860.284934

[10] S. Shalev, S. Shammah, A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," Oct. 2016.
DOI: https://doi.org/10.48550/arXiv.1610.03295

[11] R. Lowe, Y. Wu, A. Tamar, et al., "Multi-agent actor-critic for mixed cooperative-competitive environmen ts," 2017.
DOI: https://doi.org/10.48550/arXiv.1706.02275

[12] T. Rashid, M. Samvelyan, C. Schroeder, et al., "Monotonic value function factorisation for deep multi-agent reinforcement learning," Aug. 2020.
DOI: https://doi.org/10.48550/arXiv.2003.08839

[13] R. Sutton, D. McAllester, S. Singh, et al., "Policy Gradient methods for Reinforcement Learning with Fu nction Approximation," 1999.

[14] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel,"Trust Region Policy Optimization," 2015. DOI: https://doi.org/10.48550/arXiv.1502.05477

[15] R. Sutton and A. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.
ISBN: 978-0262039246

[16] A. Nair, P. Srinivasan, S. Blackwell, et al., " Massively Parallel Methods for Deep Reinforcement Learning," Jul. 2015.
DOI: https://doi.org/10.48550/arXiv.1507.04296

[17] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," 2021.

[18] A. Nichol, V. Pfau, C. Hesse, O. Klimov, and J. Schulman, "Gotta Learn Fast: A New Benchmark for Generalization in RL," 2018.
DOI: https://doi.org/10.48550/arXiv.1804.03720