iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** II **Month of publication:** February 2024

DOI: https://doi.org/10.22214/ijraset.2024.58270

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

Decision Making in Monopoly Using a Hybrid Deep Reinforcement Learning Approach

Hitesh Parihar[1], Suman M[2]
B.Tech in Computer Science & Engineering, Dept of CSE, Presidency University

Abstract: ***This research introduces a novel approach to decision-making in the classic board game Monopoly by leveraging a hybrid deep reinforcement learning framework. Traditional methods for decision-making in Monopoly often rely on heuristic strategies or rule-based systems, which may lack adaptability and fail to capture the dynamic and complex nature of the game. In contrast, this study proposes a hybrid model that combines the strengths of deep learning and reinforcement learning to enable an agent to autonomously learn effective decision-making strategies through interactions with the Monopoly environment.***

The hybrid deep reinforcement learning model is designed to process and interpret the intricate game state representations, learning optimal decision policies over time. The incorporation of deep neural networks allows the model to capture complex patterns and dependencies in the data, while reinforcement learning enables the agent to iteratively improve its decision-making abilities through trial and error. The research focuses on developing a suitable reward structure and feature representation that enhances the model's ability to navigate the diverse scenarios encountered in Monopoly, ultimately aiming to achieve superior decision-making performance compared to traditional approaches.

Through comprehensive experiments and evaluations, the study demonstrates the efficacy of the proposed hybrid approach in Monopoly decision-making. Results showcase the model's capacity to adapt to various game dynamics and outperform baseline methods. The findings not only contribute to advancing decision-making techniques in board games but also hold broader implications for the application of hybrid deep reinforcement learning in real-world scenarios with complex decision spaces.

Keywords: Monopoly, Decision-making, Hybrid model, Deep reinforcement learning, Board games, Game state representation, Neural networks, Reinforcement learning.

I. INTRODUCTION

Monopoly, an enduring classic in the realm of board games, has long captured the fascination of players worldwide with its blend of chance, strategy, and negotiation.

The game's complexity arises from the myriad decisions players must make throughout, involving property acquisition, resource management, and adaptive strategies in response to opponents' actions. Traditional approaches to decision-making in Monopoly have largely relied on heuristics and rule-based systems, providing predetermined strategies that may falter in the face of dynamic and evolving game scenarios. In response to this limitation, this research proposes an innovative solution leveraging a hybrid deep reinforcement learning approach to enhance decision-making processes within the Monopoly framework.

The motivation for employing a hybrid model arises from the recognition that the intricacies of Monopoly demand a more adaptive and learning-centric approach.

Deep reinforcement learning, a synergy of deep neural networks and reinforcement learning, holds promise in addressing the challenges posed by the game's dynamic nature. By incorporating deep learning, the model gains the ability to discern intricate patterns within the game state representations, while reinforcement learning empowers the agent to learn and refine decision strategies through interactions with the Monopoly environment. This amalgamation aims to create a sophisticated decision-making agent capable of adapting to diverse and unpredictable game states, transcending the limitations of static rule-based systems. The objectives of this study extend beyond theoretical advancements, focusing on the practical implementation of a hybrid deep reinforcement learning model. Central to this effort is the development of an effective reward structure and feature representation that enables the model to interpret and respond to the nuances of the Monopoly game. Through extensive experimentation and evaluation, this research seeks to showcase the superiority of the proposed hybrid approach over traditional heuristic methods, offering insights into the potential of advanced artificial intelligence techniques in transforming decision-making processes in strategic board games. Beyond the gaming domain, the implications of this research may extend to real-world applications with complex decision spaces, underscoring the broader significance of the study.

©IJRASET: All Rights are Reserved

II. OBJECTIVES

1) *Develop a Hybrid Model:* Construct a hybrid deep reinforcement learning model specifically tailored for decision-making in Monopoly. Integrate deep neural networks with reinforcement learning techniques to create an adaptive agent capable of learning optimal strategies through interactions with the game environment.
2) *Enhance Game State Representation:* Design and implement an effective feature representation for the Monopoly game state. Explore how the hybrid model can capture and interpret complex patterns within the game, allowing the agent to make informed decisions based on a nuanced understanding of the evolving board state.
3) *Formulate an Optimal Reward Structure:* Develop a reward structure that encourages the hybrid model to make decisions aligning with strategic objectives in Monopoly. Investigate the challenges of designing a reward system that balances short-term gains with long-term strategic goals, enhancing the model's ability to learn and adapt.
4) *Enable Autonomous Learning:* Implement mechanisms within the hybrid model to enable autonomous learning. Design the training procedures to allow the agent to iteratively refine its decision-making strategies through trial and error, adapting to different player behaviors and varying game dynamics.
5) *Compare with Baseline Methods:* Conduct comprehensive experiments to evaluate the performance of the hybrid deep reinforcement learning model. Compare its decision-making capabilities against baseline methods, such as traditional rule-based systems and heuristic strategies, to quantify the improvement achieved by the proposed approach.
6) *Analyze Decision Policies:* Investigate and analyze the decision policies learned by the hybrid model. Gain insights into the adaptive strategies developed during training, understanding how the model navigates complex decision spaces and responds to diverse scenarios encountered in Monopoly gameplay.
7) *Demonstrate Efficacy in Varied Scenarios:* Test the robustness of the hybrid model by exposing it to diverse Monopoly scenarios. Evaluate its performance in different game setups, considering variations in player strategies, initial board conditions, and external factors, to demonstrate the model's adaptability and versatility.
8) *Explore Real-world Implications:* Consider the broader implications of the research beyond the gaming domain. Discuss how the insights gained from the hybrid model's decision-making capabilities in Monopoly could be extrapolated to real-world scenarios with complex decision spaces, offering potential applications in areas such as strategy optimization and autonomous decision support.
9) *Contribute to Decision-making Literature:* Contribute to the academic literature on decision-making in strategic board games by providing a comprehensive exploration of the hybrid deep reinforcement learning approach. Share insights into the challenges, innovations, and potential advancements in the field, paving the way for future research and applications.

III. LIMITATIONS

Despite the promising potential of a hybrid deep reinforcement learning approach for decision-making in Monopoly, several limitations should be acknowledged. One significant challenge lies in the complexity of reward design. Crafting a reward structure that appropriately balances short-term gains and long-term strategic goals is a non-trivial task, and the model's performance may be sensitive to the specific choices made in this regard. A suboptimal reward design could result in the model learning strategies that prioritize immediate gains at the expense of overarching game objectives. Another noteworthy limitation is the data efficiency and training time required. Training a deep reinforcement learning model in the context of a board game like Monopoly demands substantial computational resources and time. The extended training periods may hinder the practical application of the model, especially in real-time decision-making scenarios. This limitation raises concerns about the scalability and feasibility of deploying the proposed approach in situations where timely decisions are crucial. Furthermore, the generalization of the hybrid model to diverse player strategies poses a potential challenge. If the training dataset is not sufficiently diverse or representative of the full spectrum of player behaviors, the model may struggle to generalize effectively. This limitation could result in suboptimal decision-making when confronted with novel player tactics or unexpected variations in game dynamics. Addressing these limitations is essential to ensure the robustness and real-world applicability of the hybrid deep reinforcement learning approach in the nuanced and evolving landscape of Monopoly gameplay.

IV. LITERATURE SURVEY

In [1] "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm" by Silver et al. (2017), the authors present a groundbreaking exploration of deep reinforcement learning's capabilities in mastering complex board games, specifically focusing on chess and shogi. The study introduces a general reinforcement learning algorithm, showcasing its effectiveness in achieving superhuman performance through self-play.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

While the primary emphasis is on chess and shogi, the success of this algorithm lays the foundation for subsequent research endeavors exploring the application of similar techniques in strategic board games. The work not only demonstrates the adaptability and scalability of deep reinforcement learning but also inspires subsequent studies, including those investigating decision-making processes in other strategic games, such as Monopoly, leveraging the insights gained from this pioneering research.

In [2] "Heuristic Search for Monopoly Playing Agents" by Szita and Lorincz (2009), the authors contribute to the understanding of decision-making strategies in Monopoly by exploring heuristic algorithms. Focusing on the challenge of balancing risk and reward in property acquisition within the game, the study delves into traditional methods for developing Monopoly playing agents. While the work provides valuable insights into early approaches for decision-making in Monopoly, it also underscores the limitations of heuristic strategies in adapting to the dynamic and evolving nature of the game. This research laid a foundational understanding for subsequent studies in the field, prompting further exploration into more sophisticated and adaptive approaches, such as deep reinforcement learning, to address the complexities inherent in strategic board games like Monopoly.

In [3] "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play" by Silver et al. (2018), the authors build upon their prior successes and extend their exploration of deep reinforcement learning to master a diverse set of complex board games, including chess, shogi, and Go. This study represents a significant advancement in the application of reinforcement learning to strategic games, showcasing the algorithm's adaptability and generalization across different game domains. While the primary focus is on chess, shogi, and Go, the research serves as a pivotal contribution to the broader field of artificial intelligence and game-playing agents. The success of this general reinforcement learning algorithm motivates subsequent research efforts, influencing studies exploring similar techniques in strategic board games like Monopoly and inspiring a shift towards more adaptable and generalized approaches to decision-making in game environments.

In [4] "Hybrid Deep Reinforcement Learning: Integrating Symbolic Reasoning and Neural Networks for Cognitive Architectures" by Zhang et al. (2021), the authors contribute to the advancement of cognitive architectures by exploring a novel hybrid approach that integrates symbolic reasoning with neural networks. This study addresses the limitations of purely data-driven models by incorporating explicit symbolic knowledge, thereby enhancing the interpretability and reasoning capabilities of the resulting cognitive system. While the primary focus is on broader applications, the implications of this hybrid approach extend to decision-making in complex environments, providing a potential avenue for improving adaptability and robustness. This work contributes valuable insights into the synergy between symbolic reasoning and neural networks, laying the groundwork for innovative approaches that could be applicable to diverse domains, including strategic board games like Monopoly.

In [5] "Deep Reinforcement Learning in Board Games" by Silver et al. (2016), the authors provide a comprehensive exploration of the application of deep reinforcement learning to various board games, including chess, shogi, and Go. This work marks a significant milestone in the intersection of artificial intelligence and strategic games, demonstrating the adaptability and scalability of deep reinforcement learning techniques in mastering complex game environments. While Monopoly is not explicitly addressed, the study contributes valuable insights into the challenges and opportunities presented by the application of deep reinforcement learning to decision-making processes in strategic board games. The success demonstrated in this study laid the foundation for subsequent research endeavors, inspiring investigations into adapting similar techniques to the nuanced dynamics of games like Monopoly, where chance, strategy, and adaptability are key elements.

In [6] "Monte Carlo Tree Search in Poker using Recurrent Neural Networks" by Moravčík et al. (2017), the authors contribute to the exploration of decision-making in poker by incorporating Monte Carlo Tree Search (MCTS) with recurrent neural networks. This study showcases the integration of advanced computational techniques to address the challenges of imperfect information games, a category to which poker belongs. While the primary focus is on poker, the research provides insights into the synergy between MCTS and recurrent neural networks, offering a potential framework for enhancing decision-making strategies in complex and uncertain game environments. The study has implications for strategic board games, including Monopoly, by inspiring a deeper understanding of how machine learning techniques can be adapted to handle the intricacies of decision-making in scenarios where incomplete information plays a crucial role.

In [7] "DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker" by Moravčík et al. (2017), the authors present DeepStack, an advanced artificial intelligence system designed for expert-level play in heads-up no-limit poker. This work addresses the complexities of poker, particularly in the context of imperfect information games, by integrating deep reinforcement learning with a strategy based on a deep neural network. While the primary focus is on poker, the study contributes valuable insights into decision-making in strategic games where uncertainty and incomplete information are inherent, offering potential applications in broader gaming domains.

The success of DeepStack serves as an inspiration for adapting similar techniques to other strategic board games like Monopoly, motivating subsequent research endeavors to explore the application of deep reinforcement learning in handling the nuanced decision spaces of complex board games.

In [8] "Mastering Chess and Shogi by Planning with a Learned Model" by Silver et al. (2018), the authors extend their exploration of deep reinforcement learning to include planning with a learned model in the context of chess and shogi. This work introduces the idea of leveraging a learned model to simulate potential future states and inform decision-making strategies. While Monopoly is not explicitly addressed, the study contributes to the broader understanding of incorporating learned models into decision-making processes in strategic games. The combination of deep reinforcement learning with planning using a learned model represents a significant advancement in addressing the complexities of chess and shogi gameplay, inspiring further research into how similar techniques might enhance decision-making in other strategic board games like Monopoly, where forward planning and adaptability are key elements.

In [9] "Playing Atari with Deep Reinforcement Learning" by Mnih et al. (2013), the authors present a pioneering work that applies deep reinforcement learning to playing a variety of Atari 2600 games. This study demonstrates the adaptability of deep Q-networks (DQN) in learning directly from raw visual input, showcasing significant improvements in game-playing performance across a diverse set of Atari games. While the primary focus is on video games, the success of DQN in mastering a range of challenging environments has profound implications for decision-making in strategic board games. This research serves as a catalyst for subsequent studies, inspiring investigations into how deep reinforcement learning techniques might be tailored to handle the complexities of decision-making processes in games like Monopoly, where strategic planning and adaptability are critical components.

In [10] "AlphaGo: Mastering the Ancient Game of Go with Machine Learning" by Silver et al. (2016), the authors present a groundbreaking achievement in artificial intelligence by introducing AlphaGo, a system that mastered the ancient and highly complex game of Go. Utilizing a combination of deep neural networks and reinforcement learning, AlphaGo demonstrated superhuman performance, defeating world champion players. While the primary focus is on Go, a board game with immense strategic complexity, the success of AlphaGo has far-reaching implications for decision-making in strategic board games. The study has inspired subsequent research endeavors, encouraging investigations into the application of machine learning techniques in understanding and enhancing decision-making processes in other complex games such as Monopoly. The work represents a pivotal moment in the intersection of artificial intelligence and strategic gameplay, highlighting the potential for advanced machine learning methods to tackle challenges in diverse decision spaces.

V. BACKGROUND

A. Monopoly

Monopoly is a classic board game where players engage in strategic decision-making based on their position on the game board. The game board comprises 40 square locations, as depicted in Figure 1. These locations include various properties, railroads, utilities, tax spots, card draws, jail, and other key landmarks. The 40-square game board encompasses 28 property locations, organized into eight color groups, with 22 real-estate properties. Additionally, there are four railroads, two utility spots, two tax locations imposing fees upon landing, six card spots requiring players to draw from community chest or chance card decks, as well as the jail, go to jail, go, and free parking spots.

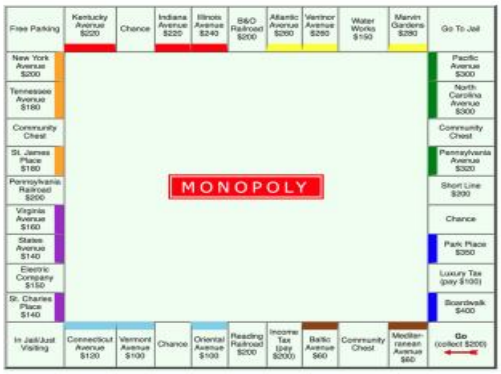


Fig. 1: Monopoly game board

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

Each property on the game board is associated with a purchase price, rent value, and color, as illustrated in the Figure 1 Players act as property owners, aiming to acquire, trade, and manage these properties effectively. The objective is to strategically buy, sell, and improve properties to accumulate wealth and drive opponents into bankruptcy.
In essence, Monopoly simulates real estate investment and management, challenging players to make astute financial decisions and outmaneuver opponents. The winner emerges as the last player standing, having outsmarted others through shrewd property transactions and strategic gameplay.

B. Markov Decision Process

An MDP (Markov Decision Process) is characterized by a tuple denoted as $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where:

- $\mathcal{S}$ represents the set of all possible states that the system can be in.
- $\mathcal{A}$ denotes the set of all possible actions that the agent can take.
- The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ calculates the probability that executing action $\mathcal{A}$ in state $\mathcal{S}$ will lead to a transition to state s' .
- The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ defines the immediate reward received by the agent upon executing action $\mathcal{A}$, resulting in a transition from state $\mathcal{S}$ to state s' .

In simpler terms, an MDP describes the dynamics of an environment in terms of states and actions. The transition function quantifies the likelihood of transitioning from one state to another after taking a specific action, while the reward function quantifies the immediate reward associated with that transition. Together, these components provide a formal framework for modeling decision-making under uncertainty in various scenarios, including reinforcement learning and optimization problems.

C. Reinforcement Learning

Solving an MDP results in the determination of a policy, denoted as $\pi : \mathcal{S} \to \mathcal{A},$, which maps states to actions. An optimal policy, denoted as π^*, maximizes the expected sum of rewards obtained by the agent. Reinforcement Learning (RL) offers a popular approach to solving MDPs without requiring explicit specification of transition probabilities. In RL, an agent interacts with the environment over discrete time steps, gradually learning the optimal policy through trial and error.
Deep Reinforcement Learning (DRL) extends RL by employing deep neural networks to approximate either the optimal policy or the value function, addressing the limitations of traditional RL methods. By using deep neural networks as function approximators, DRL allows for powerful generalization but demands careful consideration of representation choices. Inadequate design decisions may lead to estimates diverging from the optimal policy.
Existing model-free DRL methods fall broadly into two categories: policy gradient and value-based methods. Policy gradient methods directly optimize the policy using deep networks, making them suitable for scenarios with continuous action spaces. Popular policy gradient methods include Deep Deterministic Policy Gradient (DDPG), Asynchronous Advantage Actor-Critic (A3C), Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO).
On the other hand, value-based methods focus on estimating the value of being in a given state. Deep Q-Network (DQN) stands out as a well-known value-based DRL method. Numerous extensions of the DQN algorithm have emerged in recent years, including Double DQN (DDQN), Distributed DQN, Prioritized DQN, Dueling DQN, Asynchronous DQN, and Rainbow DQN.
In our study, we implement both a policy gradient method (PPO) and a value-based method (DDQN) to train our standard DRL and hybrid agents. This approach allows us to explore the strengths and weaknesses of each method and evaluate their performance in various scenarios (refer to Section V for detailed analysis).

VI. RELATED WORK

While Monopoly enjoys widespread popularity, previous research on learning-based decision-making for the full game has been limited. Some earlier attempts have modeled Monopoly as a Markov Process [32], while more recent efforts have framed it as an MDP [9][10]. However, these endeavors focused on simplified versions of the game, restricting actions to buying, selling, or doing nothing, without considering player trades.

©IJRASET: All Rights are Reserved

For instance, in [9], researchers trained and tested an RL agent against random and fixed-policy opponents using Q-learning and a neural network. Similarly, [10] employed a feed-forward neural network with experience replay to learn gameplay. However, both studies underscored the complexity of maintaining high win rates against diverse opponent strategies.

Comparatively, Settlers of Catan, another board game involving player trades, presents similar challenges. Both games feature non-uniform action distributions, with certain actions (e.g., trades) more frequently applicable. In Monopoly, players can trade freely, whereas property acquisition is constrained to landing on unowned squares. [33] tackled Settlers of Catan using a model-based approach, MCTS, handling the skewed action space by sampling from distributions.

In other domains, hybrid DRL techniques have emerged to mitigate computational complexity and enhance decision-making. For instance, [15] combined DQN for high-level decision-making with rule-based constraints for safe autonomous driving. Similarly, [12] integrated DRL and rule-based methods for robotic object manipulation, while [11] proposed a hybrid algorithm for power operations.

In our work, we utilize DRL to address decision-making in Monopoly. Unlike prior attempts, we maintain the game's full complexity, considering all possible actions, including trades. This comprehensive approach poses challenges due to the high-dimensional action space. We enhance the state space representation and reward function for greater realism.

To manage the non-uniform action space, we propose a hybrid agent that combines fixed-policy and DRL approaches. We distinguish actions based on their frequency, leveraging rule-based decisions for rare actions and DRL for others. This approach, while existing in literature, is domain-specific, and our work is the first to base the distinction on action frequency.

Our study explicitly demonstrates the superiority of the hybrid approach over standard methods under constant conditions. Additionally, we compare policy gradient and value-based methods for our hybrid agent, providing insights into their effectiveness.

VII. MDP MODEL FOR MONOPOLY

A. State Place

We define the state as a composite of player and property representations. For the player component, we include the player's current location, cash holdings, and flags indicating whether the player is in jail or possesses a get-out-of-jail-free card. Since other cards in the game immediately trigger actions and do not affect decision-making, we omit them from the state representation.

The property component encompasses the 28 property locations on the game board, including 22 real-estate properties, four railroads, and two utilities. Each property's representation includes information about its owner, mortgage status, monopoly status, and the fraction of houses and hotels built relative to the maximum allowed.

To represent property ownership, we use a 4-dimensional one-hot-encoded vector, with each index corresponding to a player (or the bank if unowned), ensuring clarity about property ownership. Properties outside of real estate groups have zero values for house and hotel attributes, as they are not applicable.

We exclude non-property locations from the state representation, as they do not require decisions from the agent. Consequently, the overall state space representation becomes a 240-dimensional vector: 16 dimensions for player information and 224 dimensions for property details. This representation captures essential aspects of the game state, enabling effective decision-making in Monopoly.

B. Action Space

We consider all actions in Monopoly that necessitate a decision by the agent, excluding compulsory actions like paying taxes, moving due to a chance card, or paying rent. An exhaustive list of these actions can be found in Table I.

Actions in Monopoly are broadly categorized into three groups: those associated with all 28 properties, those tied to the 22 color-group properties, and those unrelated to any properties. Actions not linked to properties are represented as binary variables. Since only the "Use get out of jail card" action involves card usage, and it can be deferred, we include it as a separate action. Other cards prompt immediate actions, precluding the need for decision-making.

Property improvements (building houses or hotels) are only allowed for color-group properties. Therefore, we represent these actions as a 44-dimensional vector: 22 dimensions for building houses and 22 for building hotels on specific properties. Actions associated with all properties, except buying and making trade offers, are represented using a 28-dimensional one-hot-encoded vector.

Regarding trades, players can negotiate at any time during the game, introducing complexity. A trade offer encompasses various parameters: target player, offered property, requested property, offered cash, and requested cash. We categorize trade offers into three sub-actions: selling, buying, and exchanging properties.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

For buying/selling trade offers, we discretize cash into three tiers: below market price, at market price, and above market price. Considering three other players, 28 properties, and three cash amounts, we represent these using a 252-dimensional vector.
Exchange trade offers involve swapping properties at market price, necessitating representation of properties and players. We represent these offers using a 2268-dimensional vector.
Altogether, the action space comprises 2922 dimensions, reflecting the multitude of decisions players face during Monopoly gameplay. This comprehensive representation enables our agent to make informed decisions across various scenarios encountered in the game.

VIII. APPROACH

A. *Standard DRL Agents*

1) *Actor-Critic PPO Agent*

a) We implement the actor-critic Proximal Policy Optimization (PPO) algorithm with a clipped surrogate objective function, as described in [25].
b) To estimate the advantage, we employ a truncated version of generalized advantage estimation used in [23].
c) The actor-critic PPO approach utilizes two separate networks: the actor network and the critic network.
d) PPO operates as an on-policy algorithm, where the agent follows a policy dictated by the actor network for a fixed number of time steps, typically much less than the episode length.
e) Initially, the parameters of the actor network are randomly initialized, and the agent explores by sampling actions according to its stochastic policy.
f) As training progresses, the policy tends to become less random as the update rule incentivizes the agent to exploit rewards.
g) The critic network calculates the value of a given state, which is crucial for advantage estimation when optimizing the objective function.
h) For further details on the methodology, please refer to Appendix A-A.

2) *DDQN Agent*

a) A common issue encountered with vanilla Deep Q-Network (DQN) is the tendency to overestimate the expected return.
b) Double Q-learning [34] addresses this issue by employing a double estimator.
c) To mitigate overestimation of Q-values, we implement the Double DQN (DDQN) algorithm [27] to train our agent.
d) Similar to standard DQN, DDQN utilizes experience replay [35] and a target network.
e) We adopt an ε-greedy exploration policy to select actions, where initially, the agent explores the environment by randomly sampling from allowed actions.
f) As the learning progresses and the agent discerns more successful actions, its exploration rate diminishes in favor of exploiting learned knowledge.
g) We mask the network's output to include only allowed actions, facilitating faster training by ensuring the selection of valid actions at any given time.
h) For more comprehensive insights into the methodology, please consult Appendix A-B.

These methods represent advanced techniques in reinforcement learning, each tailored to address specific challenges encountered in training agents for complex decision-making tasks like playing Monopoly. For a deeper understanding of the implementation details and their respective advantages, readers are encouraged to refer to the appendices.

B. *Hybrid Agents*

Standard Deep Reinforcement Learning (DRL) techniques often suffer from high sample complexity, requiring each state-action pair to be visited infinitely often for effective learning. To address this challenge, we employ an ε-greedy strategy in our DDQN agent, allowing the agent to balance exploration and exploitation. However, in certain scenarios where states are rare or actions are straightforward, forcing the agent to explore them may not be efficient, especially when we already have insights into potentially favorable actions.
In Monopoly, specific actions are infrequently allowed, such as buying unowned properties or accepting trade offers. For instance, a player can only purchase a property when they directly land on its square, and accepting a trade offer is only valid when such an offer exists from another player. These rare occurrences of action-state pairs further exacerbate the sample and computational complexity of the learning task.

To mitigate these challenges and enhance overall performance, we propose a hybrid approach that integrates both DRL and fixed-policy techniques. Specifically, we use a fixed-policy approach to handle rare but straightforward decisions, such as buying properties and accepting trade offers. For all other decisions, we rely on DRL methods.

During training, if there is an outstanding trade offer, the decision-making process shifts from the learning-based agent to a fixed-policy agent to determine whether to accept the offer. The agent will accept a trade offer if it leads to an increase in the number of monopolies owned, as this is generally considered advantageous in Monopoly.

This hybrid agent architecture leverages the strengths of both approaches, utilizing rule-based strategies for simple decisions and learning-based methods for more complex scenarios. By combining these techniques, we aim to improve the efficiency and effectiveness of decision-making in Monopoly gameplay.

IX. EXPERIMENTAL SETUP

A. *Monopoly Simulator*

We've developed an open-source simulator for a four-player game of Monopoly using Python, available on GitHub. This simulator replicates the conventional Monopoly board with its 40 locations, as depicted in Figure 1, and enforces rules similar to the US version of the game, with some minor modifications. Notably, we do not incorporate rules associated with rolling doubles, treating them as standard dice rolls without special privileges such as getting out of jail.

Trading is a fundamental aspect of Monopoly gameplay. Players can negotiate trades to exchange properties with or without cash, subject to certain rules:

1) Trades involve only unimproved (no houses or hotels) and unmortgaged properties.
2) Players can make trade offers to multiple opponents simultaneously. A recipient of a trade offer has the liberty to accept or reject it. The trade transaction is finalized only upon acceptance, at which point all other simultaneous trade offers for the same property are terminated.
3) Each player can have only one outstanding trade offer at a time. They must either accept or reject a pending offer before another offer can be made.

In the conventional setting, players can take certain actions like mortgaging or improving their properties even when it's not their turn to roll the dice. However, allowing multiple players to take simultaneous actions can lead to game instability. Therefore, we divide the gameplay into three distinct phases:

a) *Pre-roll:* Before rolling the dice, the active player is allowed to take certain actions. This phase concludes when the player finishes their actions.
b) *Out-of-turn:* After the pre-roll phase ends, other players can make decisions before the active player rolls the dice. Each player, in a round-robin fashion, can take actions until all players decide to skip their turn or a predetermined number of out-of-turn rounds are completed.
c) *Post-roll:* After the active player rolls the dice, their position is updated based on the sum of the dice roll.

This structured approach to gameplay phases ensures smooth and organized gameplay, facilitating the tracking of dynamic changes throughout the game.

B. *Basline Agents*

We've developed baseline agents that incorporate buying, selling, and trading properties, drawing inspiration from successful tournament-level strategies employed by human players. While informal sources on the web have documented these strategies, there hasn't been a comprehensive academic study to determine which strategies offer the highest probabilities of winning. The complex nature of Monopoly's rules has made it challenging to formalize analytically.

Our approach involves creating three fixed-policy agents: FP-A, FP-B, and FP-C. These agents are equipped to engage in one-way (buy/sell) or two-way (exchange) trades, with or without cash involvement. They have the capability to simultaneously roll out trade offers to multiple players, thereby increasing the likelihood of successful trades and facilitating the acquisition of properties leading to monopolies within specific color groups.

The fixed-policy agents adopt a strategic approach when making trade offers, aiming to offer properties that may hold low value to themselves but could be valuable to the other player, potentially resulting in the creation of monopolies. Additionally, they prioritize improving monopolized properties by constructing houses and hotels to bolster their cash balance.

While all three agents prioritize acquiring monopolies, they differ in their property preferences and trading strategies:

1) FP-A assigns equal priority to all properties.

2) FP-B and FP-C prioritize the acquisition of the four railroad properties. FP-B also places high importance on high-rent locations like Park Place and Boardwalk, while assigning a lower priority to utility locations. Conversely, FP-C prioritizes properties in the orange or sky-blue color groups.

3) Agents may adopt aggressive strategies to acquire properties of interest, even at the expense of a low cash balance. They may also opt to sell lower-priority properties to generate cash for acquiring properties deemed more valuable.

This strategic framework enables our agents to navigate Monopoly gameplay effectively, leveraging fixed-policy approaches informed by successful player strategies to enhance their performance and strategic decision-making abilities.

X. CONCLUSION

We introduce a pioneering attempt to model the complete version of Monopoly as a Markov Decision Process (MDP), leveraging innovative state and action space representations along with an enhanced reward function. Through our approach, we demonstrate that our Deep Reinforcement Learning (DRL) agent effectively learns to outperform various fixed-policy agents.

The inherent complexity of decision-making in Monopoly, exacerbated by the non-uniform action distribution, poses a significant challenge. To address this, we propose a hybrid DRL approach, wherein DRL is employed for complex decisions occurring frequently, while a fixed-policy guides simpler, less frequent decisions.

Our experimental results showcase the superior performance of hybrid agents over standard DRL agents, encompassing both policy-based (Proximal Policy Optimization, PPO) and value-based (Double Deep Q-Network, DDQN) methods. Across diverse scenarios, hybrid agents consistently achieve high win rates. Notably, the hybrid PPO agent emerges as the most effective, boasting a remarkable 91% win rate against baseline agents and a 69.06% win rate against other learning agents.

While we integrate a fixed-policy approach with DRL in this study, we recognize the potential for exploring alternative hybrid approaches. For instance, instead of employing a fixed-policy agent for infrequent actions, a separate learning agent could be tasked with handling such actions, either jointly or independently trained from the primary learning agent.

In future endeavors, we aim to explore additional hybrid approaches, delve into Multi-Agent Reinforcement Learning (MARL) techniques to train multiple agents, and enhance the Monopoly simulator to accommodate human opponents. This ongoing exploration promises to further advance our understanding and optimization of decision-making strategies in Monopoly gameplay.

REFERENCES

[1] Celli, A. Marchesi, T. Bianchi, and N. Gatti, “Learning to correlate in multi-player general-sum sequential games,” in Advances in Neural Information Processing Systems, 2019, pp. 13 076–13 086.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al., “Mastering the game of go with deep neural networks and tree search,” nature, vol. 529, no. 7587, pp. 484–489, 2016.

[3] I. Oh, S. Rho, S. Moon, S. Son, H. Lee, and J. Chung, “Creating pro level ai for a real-time fighting game using deep reinforcement learning,” IEEE Transactions on Games, 2021.

[4] P. Paquette, Y. Lu, S. S. Bocco, M. Smith, O. G. Satya, J. K. Kummer feld, J. Pineau, S. Singh, and A. C. Courville, “No press diplomacy: Modeling multi agent gameplay,” in Advances in Neural Information Processing Systems, 2019, p. 4474 4485.

[5] M. Morav˘c´ık, M. Schmid, N. Burch, V. Lis`y, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling, “Deepstack: Expert level artificial intelligence in heads up no limit poker,” Science, vol. 356, no. 6337, p. 508 513, 2017.

[6] N. Brown and T. Sandholm, “Superhuman ai for multiplayer poker,” Science, vol. 365, no. 6456, p. 885 890, 2019.

[7] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev et al., “Grand master level in starcraft ii using multi agent reinforcement learning,” Nature, vol. 575, no. 7782, p. 350 354, 2019.

[8] K. Shao, Y. Zhu, and D. Zhao, “Starcraft micromanagement with rein forcement learning and curriculum transfer learning,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, no. 1, pp. 73– 84, 2019.

[9] P. Bailis, A. Fachantidis, and I. Vlahavas, “Learning to play monopoly: A reinforcement learning approach,” in Proceedings of the 50th Anniver sary Convention of The Society for the Study of Artificial Intelligence and Simulation of Behaviour. AISB, 2014.

[10] E. Arun, H. Rajesh, D. Chakrabarti, H. Cherala, and K. George, “Monopoly using reinforcement learning,” in TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019, pp. 858–862.

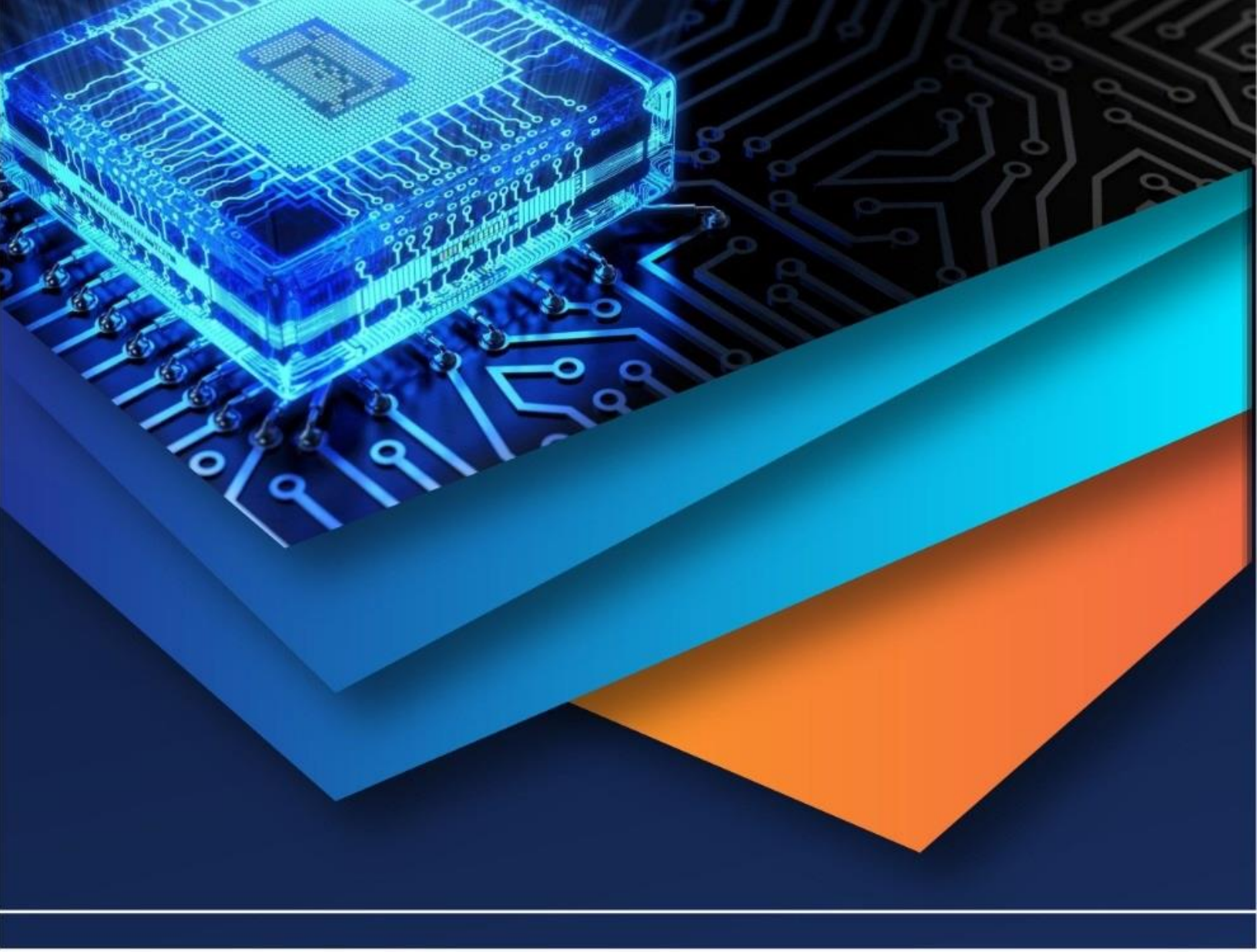
[11] D. Lee, A. M. Arigi, and J. Kim, “Algorithm for autonomous power increase operation using deep reinforcement learning and a rule-based system,” IEEE Access, vol. 8, pp. 196 727–196 746, 2020.

[12] Y. Chen, Z. Ju, and C. Yang, “Combining reinforcement learning and rule-based method to manipulate objects in clutter,” in 2020 Interna tional Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–6.

[13] H. Xiong, T. Ma, L. Zhang, and X. Diao, “Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable driven parallel robots,” Neurocomputing, vol. 377, pp. 73–84, 2020.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[14] A. Likmeta, A. M. Metelli, A. Tirinzoni, R. Giol, M. Restelli, and D. Ro mano, "Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving," Robotics and Autonomous Systems, vol. 131, p. 103568, 2020.
[15] J. Wang, Q. Zhang, D. Zhao, and Y. Chen, "Lane change decision making through deep reinforcement learning with rule-based con straints," in 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–6.
[16] Q. Guo, O. Angah, Z. Liu, and X. J. Ban, "Hybrid deep reinforce ment learning based eco-driving for low-level connected and automated vehicles along signalized corridors," Transportation Research Part C: Emerging Technologies, vol. 124, p. 102980, 2021.
[17] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.
[18] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, 2017.
[19] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in Machine Learning Proceedings 1995. Elsevier, 1995, pp. 30–37.
[20] S. Whiteson, "Evolutionary function approximation for reinforcement learning," Journal of Machine Learning Research, vol. 7, 2006.
[21] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approxi mation error in actor-critic methods," in International Conference on Machine Learning. PMLR, 2018, pp. 1587–1596.
[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
[23] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep rein forcement learning," in International conference on machine learning. PMLR, 2016, pp. 1928–1937.
[24] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in International conference on machine learning. PMLR, 2015, pp. 1889–1897.
[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Prox imal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," nature, vol. 518, no. 7540, pp. 529–533, 2015.
[27] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in Proceedings of the AAAI conference on artificial intelligence, vol. 30, no. 1, 2016.
[28] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen et al., "Mas sively parallel methods for deep reinforcement learning," arXiv preprint arXiv:1507.04296, 2015.
[29] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," arXiv preprint arXiv:1511.05952, 2015.
[30] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in International conference on machine learning. PMLR, 2016, pp. 1995– 2003.
[31] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in Thirty second AAAI conference on artificial intelligence, 2018.
[32] R. B. Ash and R. L. Bishop, "Monopoly as a markov process," Mathematics Magazine, vol. 45, no. 1, pp. 26–29, 1972.
[33] M. S. Dobre and A. Lascarides, "Exploiting action categories in learning complex games," in 2017 Intelligent Systems Conference (IntelliSys). IEEE, 2017, pp. 729–737.
[34] H. Hasselt, "Double q-learning," Advances in neural information pro cessing systems, vol. 23, pp. 2613–2621, 2010.
[35] L.-J. Lin, "Self-improving reactive agents based on reinforcement learn ing, planning and teaching," Machine learning, vol. 8, no. 3-4, pp. 293– 321, 1992.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)