AN IMPROVED HIERARCHICAL DEEP REINFORCEMENT LEARNING FOR COMPLEX IMPERFECT-INFORMATION CARD GAMES

LUO QIAN

UNIVERSITI SAINS MALAYSIA

2025

AN IMPROVED HIERARCHICAL DEEP REINFORCEMENT LEARNING FOR COMPLEX IMPERFECT-INFORMATION CARD GAMES

by

LUO QIAN

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

March 2025

ACKNOWLEDGEMENT

The first person I would like to thank is my supervisor, Tien-Ping Tan. His guidance, support, and knowledge helped immensely in my pursuit of this degree. I learned a lot from working with him, not just about game theory but also about the value of rigor, persistence, and patience. There were times when I would start a meeting discouraged by recent results and end the meeting excited by future possibilities. This enthusiasm always propelled me forward and played an important role in my success.

I would like to extend my special thanks to the Dean and all staff members of the School of Computer Sciences, Universiti Sains Malaysia. Finally, there are no words to express my deep feelings for my family. Their encouragement and support were always appreciated and critically important at times.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Page

LIST OF APPENDICES

PEMBELAJARAN PENGUKUHAN MENDALAM HIERARKI YANG DIPERBAIKI UNTUK PERMAINAN KAD MAKLUMAT TIDAK SEMPURNA DAN KOMPLEKS

ABSTRAK

Pembelajaran pengukuhan mendalam (DRL) telah mencapai kejayaan besar dalam pelbagai permainan, sama ada dengan maklumat sempurna atau tidak sempurna, seperti Go, Texas Hold'em, dan StarCraft II. Walau bagaimanapun, DouDiZhu dan Big2, permainan kad klasik yang kompleks dan maklumat tidak sempurna yang sangat popular di Asia, mempersembahkan cabaran baru bagi Kecerdasan Buatan (AI) dari segi persaingan, kerjasama, meneka maklumat yang tidak sempurna, mengendalikan ruang tindakan yang besar, dan latihan dengan ganjaran yang jarang. Kaedah Deep Monte Carlo (DMC) untuk permainan kad ini mencapai kejayaan yang signifikan tetapi masih menghadapi tiga masalah penyelidikan utama: kelajuan pembelajaran yang perlahan, kerugian yang tinggi semasa pembelajaran, dan pengoptimuman prestasi. Objektif utama penyelidikan ini adalah untuk meningkatkan prestasi AI untuk permainan kad maklumat tidak sempurna yang kompleks ini dengan rangka kerja Pembelajaran Pengukuhan Mendalam Hierarki (HDRL). Khususnya, matlamat utama ini dibahagikan kepada tiga objektif penyelidikan kecil: meningkatkan kecekapan pembelajaran dalam latihan DMC melalui Pembimbingan Oracle Guiding, meningkatkan kestabilan pembelajaran dengan Monte Carlo Mendalam Adaptif (ADMC), dan meningkatkan prestasi Pengoptimuman Dasar Proksimal (PPO) menggunakan Pembentukan Ganjaran Kelebihan Relatif (RARS). Rangka kerja HDRL mengintegrasikan Oracle Guiding, ADMC, RARS, dan PPO, membentuk proses pembuatan keputusan dua peringkat. Dengan mengabstraksi dan mempermudahkan ruang tindakan yang besar, model dasar

utama, yang dilatih dengan PPO dan RARS, membuat keputusan strategik peringkat tinggi, yang selanjutnya diperhalusi menjadi tindakan khusus oleh model sub-nilai, yang dilatih dengan Oracle Guiding dan ADMC. Oracle Guiding meningkatkan kecekapan pembelajaran dengan melatih agen Oracle menggunakan kedua-dua maklumat tidak sempurna dan sempurna, kemudian secara beransur-ansur mengurangkan pergantungan pada maklumat yang sempurna. Eksperimen ablasi menunjukkan bahawa Oracle Guiding belajar lebih cepat daripada kaedah asal dalam tempoh latihan yang sama. ADMC meningkatkan kestabilan pembelajaran dengan menggunakan kliping berat kecerunan untuk mengehadkan magnitud kemas kini, dengan itu mencegah perubahan polisi yang melampau. Hasil eksperimen menunjukkan bahawa ADMC mencapai kestabilan yang lebih baik, dengan kerugian yang lebih kecil dan kurang turun naik, berbanding kaedah asal di bawah keadaan yang sama. RARS menjana isyarat ganjaran untuk Penganggar Kelebihan Umum (GAE) dalam PPO, membolehkan latihan yang efektif dalam persekitaran ganjaran yang jarang. Hasil eksperimen menunjukkan bahawa RARS meningkatkan prestasi PPO dalam permainan kad kompleks ini dengan menyediakan ganjaran yang lebih padat. Hasil penilaian menunjukkan bahawa rangka kerja HDRL meningkatkan prestasi AI secara signifikan dalam kedua-dua DouDiZhu dan Big2. Eksperimen perbandingan dengan garis dasar terkini menunjukkan bahawa rangka kerja HDRL mengatasi model DMC terkini selepas 30 hari latihan sendiri. Penemuan ini mengesahkan keberkesanan kaedah yang dicadangkan dalam meningkatkan kecekapan pembelajaran, kestabilan, dan prestasi dalam permainan kad maklumat yang tidak sempurna yang kompleks.

AN IMPROVED HIERARCHICAL DEEP REINFORCEMENT LEARNING FOR COMPLEX IMPERFECT-INFORMATION CARD GAMES

ABSTRACT

Deep Reinforcement Learning (DRL) has achieved significant breakthroughs in a variety of games, both with perfect and imperfect information, such as Go, Texas Hold'em, and StarCraft II. However, DouDiZhu and Big2 are classic complex card games with imperfect information and are popular in Asia. They present new challenges for AI in competition, cooperation, inferring imperfect information, handling large state-action spaces, and training with sparse rewards. The Deep Monte Carlo (DMC) method for these card games achieves significant success but still faces three key research problems: slow learning speed, high loss during learning, and performance optimization. The primary objective of this research is to enhance the performance for these complex imperfect-information card games with a Hierarchical Deep Reinforcement Learning (HDRL) framework. Specifically, this main goal is divided into three sub-research objectives: improving learning efficiency in DMC training through Oracle Guiding, enhancing learning stability with Adaptive Deep Monte Carlo (ADMC), and improving the performance of Proximal Policy Optimization (PPO) using Relative Advantage Reward Shaping (RARS). The HDRL framework integrates Oracle Guiding, ADMC, RARS, and PPO, forming a two-stage decision-making process. By abstracting and simplifying the large action space, the main-policy model, trained with PPO and RARS, makes high-level strategic decisions, which are further refined into specific actions by the sub-value model, trained with Oracle Guiding and ADMC. Oracle Guiding improves learning efficiency by training an Oracle agent using both imperfect and perfect information, then gradually reducing reliance on the former. Ablation experi-

ments demonstrate that Oracle Guiding learns faster than the original method within the same training period. ADMC enhances learning stability by employing gradient weight clipping to constrain the magnitude of updates, thereby preventing extreme policy changes. Experiment results show that ADMC achieves better stability, with smaller loss and less fluctuation, than the original method under identical conditions. RARS generates reward signals for the Generalized Advantage Estimator (GAE) in PPO, enabling effective training in sparse reward environments. Experiment results show that RARS significantly improves the performance of PPO in these complex card games by providing denser rewards. Evaluation results demonstrate that the HDRL framework significantly improves AI performance in both DouDiZhu and Big2. Comparative experiments with state-of-the-art baselines show that the HDRL framework outperforms the state-of-the-art DMC model after 30 days of self-play and training. These findings confirm the effectiveness of the proposed methods in enhancing learning efficiency, stability, and performance in complex imperfect-information card games.

CHAPTER 1

INTRODUCTION

Deep Reinforcement Learning (DRL) (Sutton and Barto, 2018; Wang et al., 2022b; Yin et al., 2024) has achieved several breakthroughs in game playing. In particular, perfect information games, where the complete game state is accessible to all players, have seen machines outperforming human professional players. Examples include Atari (Mnih et al., 2015), Chess (Silver et al., 2017), Go (Schrittwieser et al., 2020), and Tic-Tac-Toe (van Opheusden et al., 2023). However, in some imperfect information games, where some information is hidden from players, humans still have an advantage over state-of-the-art agents. Recent advances, though, show that this gap is narrowing, as seen in real-time strategy game StarCraft II (Vinyals et al., 2019) and board game Stratego (Perolat et al., 2022).

Complex imperfect-information card games are a subset of imperfect information games. They are valuable not only because they attract millions of players daily but also because they reflect numerous real-world scenarios, such as negotiations, surgical operations, business, and physics. Games provide a virtual world for Artificial Intelligence (AI) to explore, mirroring real-life problems like real-time decision making, scheduling, and education (Hu et al., 2024). Such games serve as perfect testbeds to verify the ability, generality, robustness, and safety of AI. This allows the methods and strategies refined through the study of these card games to be applied effectively in various practical contexts.

Despite the potential, training state-of-the-art models for complex card games like DouDiZhu and Big2 remains challenging due to their inherent characteristics, including imperfect information with multiple players, a large and complex state and action space, and sparse rewards. This research aims to enhance the performance for these games through a Hierarchical Deep Reinforcement Learning (HDRL) framework. HDRL is a DRL method that decomposes complex tasks into multiple sub-tasks and uses a hierarchical structure for learning and decision-making (Chai et al., 2023; Gao et al., 2024; Vezhnevets et al., 2017). By breaking down intricate tasks into a series of simpler sub-tasks, with each layer focusing on solving decision problems at a specific level, it enables improved performance in complex scenarios.

1.1 Definition of Terms

This section provides definitions for key terms used throughout this research. By clearly defining these terms beforehand, it aims to ensure a consistent understanding of the concepts and to minimize potential confusion.

- **Imperfect information.** Imperfect information is when players in a game lack complete knowledge about the game's state or the actions of other players (Solinas et al., 2024). Classic Imperfect-information games include most card games, where players cannot see their opponents' cards and must make decisions based on incomplete information.

- **Perfect information.** Perfect information means all players have complete knowledge of the entire game state and the actions taken by all players at all times. In perfect-information games, there is no hidden information, and every

player can see the moves made by others and the current state of the game. Examples of perfect-information games include Chess, Shogi, and Go, where each player can see the entire board and all pieces at all times (Schrittwieser et al., 2020).

- **Agent.** An agent in Deep Reinforcement Learning (DRL) is the entity that interacts with the environment to learn optimal behaviors through trial and error (Milani et al., 2024; Sutton and Barto, 2018). It observes the current state, selects actions based on its policy, and receives rewards or penalties from the environment, which guide its future decisions. For example, in a self-driving car scenario, the agent is the AI system controlling the car. It processes sensor data to understand its surroundings, chooses actions like accelerating or braking, and learns from the outcomes (safe driving or collisions) to improve its driving strategy.

- **Model.** The model in DRL is the neural network that approximates the policy or value function (Milani et al., 2024; Sutton and Barto, 2018). This neural network outputs action probabilities or value estimates, which the agent uses to make decisions. For example, in Proximal Policy Optimization (PPO), the model is a neural network that takes a state (s) as input and outputs the probabilities of possible actions. In Deep Monte Carlo (DMC), the model is the neural network that estimates Q-values for state-action pairs.

- **State space.** The state space is the set of all possible configurations or states that a system can be in (Lu et al., 2024; Tierney, 1995). Each state in this space represents a unique configuration of the system's variables. In DouDiZhu,

the state space includes all possible game states at any given time, such as the distribution of cards among the three players, current hand, sequence of played cards, player roles (landlord or peasant), and other game-specific variables like the current number of bombs. This space is highly dynamic and can be extremely large, especially in games with many cards and complex rules.

- **Action space.** The action space is the set of all possible actions that an agent can take in a given environment (Ming et al., 2023; Sutton and Barto, 2018). Each action in this space represents a distinct move or decision that an agent can make at a particular point in time. In DouDiZhu, the action space includes all possible plays that a player can make during their turn, such as playing a single card, a pair, a straight, a bomb, or passing the turn. This space is influenced by the rules of the game, the current state of play, and the cards available in the player's hand. The action space can be vast and complex, particularly in games with many potential combinations and strategies.

- **Sparse reward.** A sparse reward in DRL is when the agent receives feedback (rewards) infrequently or only under specific conditions (Wu and Chen, 2024). Unlike dense rewards, where the agent frequently receives positive or negative feedback based on its actions, sparse rewards provide limited feedback, often only when the agent achieves a significant milestone or completes the task. In DouDiZhu and Big2, a sparse reward scenario means the agent only receives a reward at the end of the game based on whether it won or lost, rather than receiving feedback after each move. This type of reward makes learning more challenging because the agent has fewer signals to guide its actions and must rely more on exploring the state space to find successful action sequences.

1.2 Research Motivation

The motivation for researching complex imperfect-information card games includes several key aspects. Firstly, these games provide a challenging environment to explore large state and action spaces and understand decision-making in uncertain scenarios (Bekius et al., 2022; Rumeser and Emsley, 2019; Świechowski and Tajmajer, 2021). Secondly, developing AI agents that excel in these games pushes the boundaries of AI capabilities in decision-making under uncertainty. These games serve as benchmarks for evaluating and comparing AI algorithms, driving improvements and advancing the field (Schmid et al., 2023; Solinas et al., 2023). Thirdly, the significant challenges that these games present for AI motivate the creation of new techniques to handle complexity, uncertainty, and large state-action spaces (Yang et al., 2023). Lastly, insights from studying these games can be applied to finance, cybersecurity, negotiation strategies, and military planning, demonstrating the research's broader impact (Henderson, 2021).

1.3 Problem Statement

Deep Reinforcement Learning (DRL) is an algorithm that combines Deep Neural Networks (DNNs) and Reinforcement Learning (RL) (Oroojlooy and Hajinezhad, 2023; Sutton and Barto, 2018; Yin et al., 2024). Among the most prominent DRL algorithms are Deep Q-Network (DQN) (Mnih et al., 2015; Nikpour et al., 2024), Asynchronous Advantage Actor-Critic (A3C) (Cho et al., 2024; Mnih et al., 2016), and Proximal Policy Optimization (PPO) (Schulman et al., 2017; Son et al., 2024). These algorithms achieve superior results in small-scale (less than 20 actions) Atari games. However, DouDiZhu and Big2 have more than 10^4 actions, in comparison to Atari. Consequently, when

DQN, A3C, and PPO are applied to these games, they may fail to converge due to the large number of output actions. Specifically, when the PPO method is applied to Big2, it only manages to defeat amateur players (Charlesworth, 2018). Meanwhile, Yang et al. (Yang et al., 2022a) and Zha et al. (Zha et al., 2021b) find that PPO, DQN, and A3C algorithms all struggle to achieve competitive performance in DouDiZhu. You et al. (You et al., 2020) propose Combination Deep Q-Learning (CQL), a two-phase DQN algorithm that decouples actions into decomposition selection and final move selection. CQL outperforms DQN and A3C in DouDiZhu (You et al., 2020). However, another study shows that CQL suffers from overestimation bias and has no advantage over rule-based heuristics AI (Zha et al., 2021b). To eliminate overestimation bias, DouZero (Zha et al., 2021b) adopts Deep Monte Carlo (DMC), a type of DRL algorithm that combines the conventional Monte Carlo method with DNNs. In a Monte Carlo self-play framework, DNNs first estimate the value of each action (Q-value), then select the action with the highest Q-value as a final move. DouZero is the forefront AI system in DouDiZhu for its remarkable performance in contrast to prior works. However, there are three challenges in DRL in complex imperfect-information card games:

- **Learning Efficiency:** Complex card games are multiplayer games with imperfect information where players have private, hidden cards. Therefore, players must decide on actions under uncertain conditions, which are more difficult than when they have perfect information and require more iterations and time. The state-of-the-art DMC system (a type of DRL algorithm) for DouDiZhu (Zha et al., 2021b) takes four GPUs and 48 CPUs for 30 days to train the model. The resource requirements and training time pose challenges for small-scale laboratories, individual researchers, and rapid deployment. This research addresses this

by proposing Oracle Guiding to improve learning efficiency, which is the first research objective (Objective 1). Oracle Guiding refers to using information that is unavailable during execution but accessible during training. In this approach, agents are initially trained with all information, gradually reducing the amount of hidden information as the training progresses.

- **Learning Stability:** Complex card games have a large and complex state and action space. With the combination of cards and the complex rules, DouDiZhu has approximately 10^{83} potential states and 27,472 possible actions, while Big2 has about 10^{28} states and 17,859 possible legal actions. This complex state and action space can lead to instability in neural network models due to increased complexity, data sparsity making some actions hard to estimate, and a higher risk of overfitting. The learning process of DMC (a type of DRL algorithm) in DouDiZhu and Big2 exhibits high and fluctuating loss (He, 2022; Wang et al., 2023a). When a learning process has high and fluctuating loss, it means the model is highly sensitive to small variations in the training data. Keeping low and stable loss ensures the training process is robust to these perturbations, allowing the agent to learn reliable policies (Hansen et al., 2021). Thus, this presents a new challenge in maintaining low and stable loss to improve training stability in DMC. This research tackles this issue by introducing Adaptive Deep Monte Carlo (ADMC) to enhance learning stability (low and stable loss), which is the second research objective (Objective 2).

- **Performance Improvement:** Complex card game environments typically have sparse rewards, with no instant rewards until the game is over. This scarcity of rewards makes performance and exploration challenging for some DRL methods.

Researchers have been striving to improve the performance of AI in imperfect-information card games, using methods ranging from rule-based approaches to tree search algorithms and DRL techniques (Brown and Sandholm, 2019; He, 2022). Despite breakthroughs in other games like Go, Texas Hold'em, and StarCraft II (Souchleris et al., 2023), DRL still faces challenges in handling the large state-action spaces and sparse rewards of complex card games like DouDiZhu and Big2. Methods like DouZero+ (Zhao et al., 2023b) and WagerWin (Wang et al., 2023a), based on DouZero, show only small improvements. This research first improves the performance by incorporating Relative Advantage Reward Shaping (RARS) to generate reward signals for the Generalized Advantage Estimator (GAE) (Jin et al., 2024; Schulman et al., 2015b) in PPO (a type of DRL algorithm), facilitating effective training in sparse reward environments (Objective 3). RARS is a key component of the Hierarchical Deep Reinforcement Learning (HDRL) framework, which also integrates Oracle Guiding and ADMC to address the performance challenge.

By addressing these challenges, the primary objective of this research is to enhance the performance for complex imperfect-information card games through the integration of Oracle Guiding, ADMC, and RARS into a HDRL framework.

1.4 Research Questions

The research questions of this research can be summarized as s follow:

- How to improve learning efficiency (reducing the training time) in Deep Monte

Carlo (DMC) training?

- How to improve the learning stability (keeping low and stable loss) in Deep Monte Carlo (DMC) training?

- How to improve the performance (winning score) for complex card games with imperfect information?

1.5 Research Objectives

The primary objective of this research is to enhance the performance for complex imperfect-information card games by integrating proposed methods into a Hierarchical Deep Reinforcement Learning (HDRL) framework. To achieve this main goal, the research is broken down into the following sub-research objectives:

- To improve learning efficiency in Deep Monte Carlo (DMC) training by introducing Oracle Guiding, which accelerates the training process. In Oracle Guiding, The Oracle agent initially trains with full information and gradually transitions to a standard agent by incrementally discarding imperfect information (discussed in Section 3.3 in Chapter 3).

- To improve learning stability in Deep Monte Carlo (DMC) training by proposing the Adaptive Deep Monte Carlo Method (ADMC), which incorporates Q-value weight clipping to prevent drastic policy changes and reduce high variance during training (discussed in Section 3.4 in Chapter 3).

- To improve the performance of Proximal Policy Optimization (PPO) in complex imperfect-information card games by designing Relative Advantage Reward

Shaping (RARS), which generates instant rewards for PPO to effectively learn in sparse reward environments (discussed in Section 3.5 in Chapter 3).

Integrating Oracle Guiding, ADMC, and RARS into the HDRL framework aims to further enhances the performance in complex imperfect-information card games, thereby achieving the primary objective of this research (discussed in Section 3.6 in Chapter 3).

1.6 Research Scope

This study focuses on complex imperfect-information card games, using DouDiZhu and Big2 as case studies detailed in B.1 and B.2. DouDiZhu and Big2 are valuable not only these games are two typical representatives of imperfect-information card games but also they are extremely popular in China and Southeast Asia, attracting millions of players daily. They have three distinctive features that attract researchers. First, DouDiZhu and Big2 are imperfect-information games with multiple players. Players can only see their own cards and the public cards; they cannot see their opponents' hands. Second, DouDiZhu and Big2 have a large and complex state and action space. Due to the combination of cards and the complex rules, DouDiZhu has approximately 10^{83} potential states and 27,472 possible actions, while Big2 has about 10^{28} states and 17,859 possible legal actions. Third, DouDiZhu and Big2 are games with sparse rewards. There are no instant rewards in each round of the confrontation between different camps until the game is over.

By exploring the specific research scopes of DouDiZhu and Big2, this study ad-

vances the understanding of these complex imperfect-information card games. It develops effective algorithms and creates tailored AI agents for each game. Furthermore, it evaluates performance and analyzes agent behavior in both games.

1.7 Research Contribution

There are three research contributions in this research. First, this research proposes Oracle Guiding to improve the learning efficiency of Deep Monte Carlo (DMC) for complex imperfect-information card games. In Oracle Guiding, the Oracle agent is initially trained using both imperfect information (the private cards of other players) and perfect information (its own cards and public cards). During training, the Oracle agent transitions to using only its own cards and public cards, unlike a standard agent using only perfect information. Ablation experiments demonstrate that Oracle Guiding learns faster than the original method on the same server within the same training time.

The second research contribution is the proposal of the Adaptive Deep Monte Carlo Method (ADMC). To mitigate high and fluctuating loss during DMC training, gradient weight clipping is introduced in ADMC to ensure moderate policy updates and prevent drastic changes. Ablation experiments demonstrate that ADMC achieves better stability than the original method. The combination of Oracle Guiding and ADMC, named OADMCDou for DouDiZhu and OADMCBig for Big2, shows improved performance and serves as a sub-value model in a Hierarchical Deep Reinforcement Learning (HDRL) framework.

Lastly, the third research contribution is the improvement in performance for complex imperfect-information card games through the integration of Relative Advantage

Reward Shaping (RARS), Oracle Guiding, and ADMC into the HDRL framework. In this framework, the main-policy model, trained with PPO and RARS, makes high-level strategic decisions that are further refined into specific actions by the sub-value model, trained with Oracle Guiding and ADMC. Experiments comparing this framework with state-of-the-art benchmarks for DouDiZhu and Big2 highlight its significant performance improvements.

1.8 Thesis Organization

The thesis is structured into five chapters, each building on the previous to enhance the performance in complex imperfect-information card games through a Hierarchical Deep Reinforcement Learning (HDRL) framework.

Chapter 1: Introduction. Provides an overview of the significance and challenges of applying Artificial Intelligence (AI) to complex imperfect-information card games. It explains the breakthroughs of Deep Reinforcement Learning (DRL) in game playing, especially in perfect information games, and highlights the difficulties DRL faces in imperfect information card games. This chapter also defines key terms, outlines the research motivation, problem statement, research questions, and objectives, and delineates the research scope and contributions.

Chapter 2: Literature Review. Reviews the existing work related to advancements and challenges in DRL applied to complex imperfect-information card games like DouDiZhu and Big2. It covers key DRL concepts, including Monte Carlo methods, Temporal-Difference methods, Actor-Critic methods, Proximal Policy Optimization, and Hierarchical Deep Reinforcement Learning. This chap-

ter also reviews representative applications of DRL in card games, discusses techniques for enhancing efficiency, stability, and performance, and identifies current research gaps, suggesting new approaches like Oracle Guiding, Adaptive Deep Monte Carlo (ADMC), and Relative Advantage Reward Shaping (RARS).

Chapter 3: Method Design and Implementation. Explains the design and implementation of the Oracle Guiding, ADMC, and RARS methods. It addresses the research questions and achieves objectives from Chapter 1. Furthermore, Chapter 3 details how these methods are integrated into the HDRL framework to improve the performance for complex imperfect-information card games.

Chapter 4: Experiment Results and Discussion. Evaluates the methods discussed in Chapter 3, including Oracle Guiding, ADMC, RARS, and the HDRL framework. It demonstrates the advantages of these methods in terms of learning efficiency, stability, and performance through experiment validation. Additionally, case studies illustrate the practical benefits of the HDRL framework in DouDiZhu and Big2 card games. Chapter 4 confirms the achievement of the research objectives and addressing the research questions outlined in Chapter 1.

Chapter 5: Conclusion and Future Work. Summarizes the main contributions and findings, discusses limitations, and suggests future research directions to further advance the performance in complex imperfect-information card games.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Recent advancements in game Artificial Intelligence (AI) have led to the development of sophisticated systems, transitioning from rudimentary agents such as the DQN Atari (Mnih et al., 2015) to more advanced systems including AlphaZero (Silver et al., 2017), Libratus (Brown and Sandholm, 2018), OpenAI Five (Berner et al., 2019), and AlphaStar (Vinyals et al., 2019). These AI systems surpass human experts, indicating a profound enhancement in decision-making. For instance, AlphaZero outperforms professional Go players in perfect-information games (PIGs), while OpenAI Five excels in imperfect-information games (IIGs), becoming the first AI to defeat world champions in an eSports competition. PIGs are games in which players can observe all game states; in contrast, IIGs refer to games in which participants cannot see the complete information of other players. With AlphaStar and OpenAI Five defeating professional players in StarCraft II (Vinyals et al., 2019) and Dota2 (Berner et al., 2019), it is evident that current AI techniques master complex games. This proficiency is not limited to these examples. Recent AI advancements in games like Honor of Kings (Yang et al., 2022b) use methods similar to AlphaStar and OpenAI Five, indicating a shared foundation in AI approaches across various game genres. These genres span board games like Go (Schrittwieser et al., 2020), Stratego (Perolat et al., 2022), and Tic-Tac-Toe (van Opheusden et al., 2023); card games such as Heads-up No-limit Texas Hold'em (HUNL) (Wang et al., 2023b), DouDiZhu (Jiang et al., 2019), and Mahjong (Wang

et al., 2022a); first-person shooters represented by Quake III Arena (Jaderberg et al., 2019); and real-time strategy games including StarCraft II, Dota2, and Honor of Kings. The AI behind these achievements include AlphaGo (Silver et al., 2016), DeepStack (Moravčík et al., 2017), AlphaGo Zero (Silver et al., 2017), AlphaZero (Silver et al., 2018), FTW (Jaderberg et al., 2019), Suphx (Li et al., 2020), JueWu (Ye et al., 2020), Commander (Wang et al., 2021), DouZero (Zha et al., 2021b), DeepNash (Perolat et al., 2022), WagerWin (Wang et al., 2023a), Cognitive Model (van Opheusden et al., 2023) and DanZero+ (Zhao et al., 2024b). Given these advancements, especially in card games, it naturally raises the question: What are the possible challenges of current techniques in complex imperfect-information card games? What is the choice of research methodology, what are the gaps, and what is the motivation for this study? This chapter aims to review recent successful AIs in complex imperfect-information card game and tries to answer these question through a thorough analysis of current techniques.

The remainder of this chapter unfolds as follows: Section 2.2 introduces the underlying theories related to Deep Reinforcement Learning (DRL). Section 2.3 reviews representative applications of complex imperfect-information card games, focusing on previous studies about AI algorithms in DouDiZhu and Big2. Following that, Section 2.4 discusses techniques for enhancing efficiency, stability, and performance in DRL. Finally, Section 2.5 identifies research gaps and motivates the study by summarizing the advantages and limitations of existing methods and suggesting new methods to improve them.

2.2 Preliminaries

2.2.1 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) combines Deep Neural Networks (DNNs) (Krizhevsky et al., 2012; Zhao et al., 2024a) with Reinforcement Learning (RL) (Plaat et al., 2023; Sutton and Barto, 2018) to enable agents to learn complex behaviors from high-dimensional inputs. It provides the fundamental theories and methods to solve complex problems like imperfect-information card games.

2.2.1(a) Deep Neural Networks

Deep Neural Networks (DNNs) are computational models with multiple layers, inspired by the human brain's structure and function (LeCun et al., 2015). They learn complex patterns from large datasets by iteratively adjusting internal parameters. Due to the complexity and variety of states, using a simple table or other forms to record states not only consumes substantial hardware resources but also results in inefficient performance. Thus, researchers usually utilize DNNs to store the value of a state. States represent the public cards and hidden cards in imperfect information card games. In these games, Deep Reinforcement Learning (DRL) uses DNNs to approximate value functions or policies, allowing agents to learn and make decisions in complex environments by mapping states to actions or value estimates. DNNs exhibit non-linearity and scalability, making them suitable for many applications. For instance, SL (Zha et al., 2021b) for DouDiZhu and (Charlesworth, 2018) for Big2 use DNNs to approximate the value of actions in different game states.

2.2.1(b) Reinforcement Learning

Reinforcement Learning (RL) (Plaat et al., 2023; Sutton and Barto, 2018), based on Markov Decision Process (MDP), is a mathematical framework that enables agents to learn optimal decision-making strategies by interacting with dynamic environments, as shown in Figure 2.1. In RL, an agent is an entity that learns to interact with an environment to achieve some objective. Agents in Deep Reinforcement Learning (DRL) often employ Deep Neural Networks (DNNs) to approximate complex functions that map observations to actions and value estimates, as shown in Figure 2.1. In a discounted episodic MDP denoted as (S, A, γ, P, r), the agent chooses an action a_t according to the policy $\pi(a_t|s_t)$ at state s_t. The environment receives the action, produces a reward r_{t+1}, and transitions to the next state s_{t+1} according to the transition probability $P(s_{t+1}|s_t, a_t)$. In RL, agents aim to find a policy π that maximizes the expected cumulative reward, often denoted as the objective function $J(\pi)$. RL achieves this objective by iteratively updating the value function $V(s)$ or the action-value function $Q(s, a)$ using methods such as the Monte Carlo method (Rubinstein and Kroese, 2016; Shonkwiler and Mendivil, 2024), Temporal-Difference method (Ren et al., 2024; Sutton and Barto, 2018), Actor-Critic (Konda and Tsitsiklis, 1999; Padhye and Lakshmanan, 2023), and Proximal Policy Optimization (PPO) (Schulman et al., 2017; Son et al., 2024). Traditionally, Monte Carlo method, Temporal-Difference method, Actor-Critic, and PPO often use a table memory structure (tabular method) to store the value function of each state or each state-action pair. This approach is inefficient for complex problems with a large number of states due to significant memory constraints. The tabular method requires maintaining a table with an entry for each possible state or state-action pair, which becomes impractical as the state space grows. However, when these methods are

combined with DNNs, they can effectively address complex problems with huge states by using the neural networks' ability to generalize and approximate functions. The Monte Carlo method and Temporal-Difference method are fundamental approaches for estimating values in RL, with Monte Carlo methods relying on complete episodes and Temporal-Difference methods updating estimates incrementally. Actor-Critic combines value-based and policy-based methods, with the critic providing value estimates and the actor adjusting the policy. PPO based on the Actor-Critic method is an optimization algorithm that constrains policy updates to prevent large policy changes. Together, they form a series of techniques used in RL for value estimation, policy learning, and optimization.

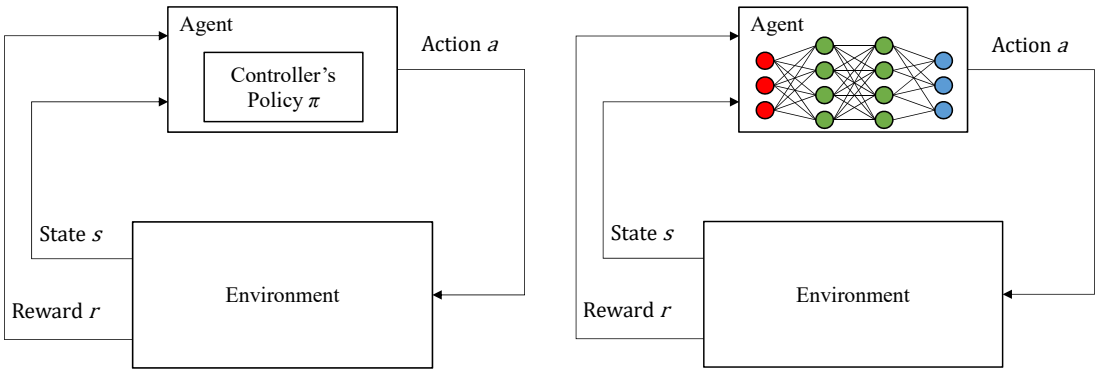

Figure 2.1: Comparison of Reinforcement Learning (left) and Deep Reinforcement Learning (DRL) (right). Agents in DRL often employ Deep Neural Networks to approximate complex functions that map observations to actions and value estimates.

Monte Carlo method. The Monte Carlo (MC) method estimates the value by repeatedly generating episodes and recording the average return at each state or each state-action pair (Binder, 2022; Luengo et al., 2020). The state value represents the expected cumulative reward starting from a specific state s_t under a given policy π, calculated as the average return from episodes starting at s_t. Similarly, the state-action

value represents the expected cumulative reward for a given state-action pair (s, a), calculated as the average return from episodes in which the agent starts at s_t and takes action a_t, as shown in Equation 2.1:

$$Q_\pi(s, a) = \lim_{k \to \infty} \mathbb{E}[r^k(s_t, a_t) | s_t = s, a_t = a, \pi] \tag{2.1}$$

Here, $r^k(s_t, a_t)$ denotes the observed return in episode k, and the expectation $\mathbb{E}$ is calculated by averaging the observed reward $r^k(s_t, a_t)$ from many episodes. The MC method does not require any knowledge of transition probabilities and is particularly suitable for episodic tasks like complex card games (Zha et al., 2021b). However, this method makes two assumptions to ensure convergence: a large number of episodes and frequent visits to every state and action. In this research, the Monte Carlo method serves as the foundation for the Deep Monte Carlo (DMC) method, which is enhanced using Oracle Guiding and ADMC to address the challenges of learning efficiency and stability in complex imperfect-information card games.

Temporal-difference method. Unlike the MC method, the Temporal-Difference (TD) method (Ren et al., 2024; Sutton and Barto, 2018) makes updates at every step within an episode rather than waiting until the end. This allows TD methods to update value estimates more frequently and potentially learn faster from fewer episodes. In TD methods, the state value or state-action value is updated at each step by considering the received reward and the estimated value of the next state or state-action pair, adjusting the current estimate based on the difference between expected and actual outcomes. The key difference between MC and TD methods lies in when they update their value

estimates. MC methods update values at the end of an episode based on total return, while TD methods update values at each time step based on expected future returns. The TD method form the foundation of the Actor-Critic method as it uses TD error to update the Critic's value estimate, guiding the Actor's policy updates.

Actor-Critic. Actor-Critic consists of value function approximation (Critic) and policy optimization (Actor), where the Actor is responsible for selecting actions and the Critic evaluates the actions taken by the Actor (Konda and Tsitsiklis, 1999; Padhye and Lakshmanan, 2023). The Critic's role is to evaluate the actions taken by the Actor by estimating the value function. To do this, the Critic uses the Temporal-Difference (TD) method, which updates the value estimates based on the TD error (δ_t). The TD error is defined as follows:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{2.2}$$

The Actor's role is to select actions based on a policy that is parameterized by θ. The Actor uses the feedback from the Critic, specifically the TD error δ_t, to update its policy parameters. The TD error δ_t represents the temporal difference in value estimates and provides a signal for how much better or worse the action a_t is compared to the expected value (Schulman et al., 2015b; Sutton and Barto, 2018). The Actor-Critic method forms the basis for Proximal Policy Optimization (PPO), which further refines policy updates to ensure stability and improve performance.

Proximal Policy Optimization. Proximal Policy Optimization (PPO) optimizes the Actor-Critic method by introducing a way to limit the step size of policy updates, which

prevents large updates that can destabilize training (Schulman et al., 2017; Son et al., 2024). Specifically, PPO uses the probability ratio between the current policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{2.3}$$

To constrain policy updates within a small range, PPO employs a clipped objective function. The clipped objective ensures that the probability ratio $r_t(\theta)$ remains within a specified range, typically $[1-\epsilon, 1+\epsilon]$, where ϵ is a small constant controlling the step size. The objective function for PPO is defined as:

$$J^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right] \tag{2.4}$$

Here, $\text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)$ limits the value of $r_t(\theta)$ to the range $[1-\epsilon, 1+\epsilon]$, thus preventing large policy updates that could destabilize training. The advantage function A_t represents the relative benefit of taking an action at a specific time step, compared to the expected value of following the policy from that state onward.

To further improve the efficiency of the advantage function estimation, PPO incorporates Generalized Advantage Estimation (GAE) (Schulman et al., 2015b). The advantage function A_t in PPO is computed using GAE as follows:

$$A_t = \sum_{k=0}^{T-t-1} (\gamma\lambda)^k \delta_{t+k} \tag{2.5}$$

where $\delta_{t+k} = r_{t+k+1} + \gamma V(s_{t+k+1}) - V(s_{t+k})$ is the TD error, and λ (a constant between 0 and 1) is used to adjust the weights of the TD errors over various time steps, acting as a trade-off hyperparameter. GAE allows PPO to efficiently approximate the advantage

function A_t by summing the discounted TD errors over multiple time steps. The clipped objective function $J^{\text{CLIP}}(\theta)$ helps in maintaining stable policy updates by ensuring that the updates do not deviate significantly from the previous policy. By integrating the clipped objective and GAE, PPO enhances the stability and performance of the Actor-Critic method. Compared to other reinforcement learning algorithms, such as Deep Q-learning (DQN) (Mnih et al., 2015), which is prone to overestimation bias and instability during training (Zha et al., 2021b), PPO offers a more stable approach due to its clipped objective and GAE mechanism. In this research, PPO is employed to improve the performance of the main-policy model in the Hierarchical Deep Reinforcement Learning (HDRL) framework for complex imperfect-information card games.

Advantages and Limitations. To provide a clear comparison, Table 2.1 summarizes the advantages and limitations of the above-discussed Reinforcement Learning methods.

Table 2.1: Advantages and Limitations of Reinforcement Learning Methods

Reinforcement Learning	Advantages	Limitations
Monte Carlo Method (Binder, 2022; Luengo et al., 2020)	1. Suitable for episodic tasks. 2. Can estimate state values or action values. 3. Can handle environments with unknown dynamics.	1. High variance in estimates. 2. Can be slow to converge. 3. May not work well in non-episodic or continuous tasks.
Temporal-Difference method (Ren et al., 2024)	1. Efficient and can update values after each time step. 2. Suitable for online learning. 3. Can handle both episodic and continuing tasks.	1. Limited exploration capability. 2. Can suffer from high bias in estimates. 3. Sensitive to initial conditions.
Actor-Critic (Padhye and Lakshmanan, 2023)	1. Combines advantages of both policy and value-based methods. 2. Provides stable learning and faster convergence. 3. Effective in continuous action spaces.	1. Requires tuning of hyperparameters. 2. Complex architecture and training. 3. Prone to overestimation bias.
Proximal Policy Optimization (PPO) (Son et al., 2024)	1. Achieves state-of-the-art performance in various domains. 2. Stable and sample-efficient. 3. Uses a simple objective function.	1. Sensitive to hyperparameter settings. 2. Can have slow convergence in some cases. 3. May not be the best choice for every problem.

2.2.2 Hierarchical Deep Reinforcement Learning

Hierarchical Deep Reinforcement Learning (HDRL) builds on Deep Reinforcement Learning (DRL) by adding a structured hierarchy to better manage complex decision-making tasks (Chai et al., 2023; Gao et al., 2024; Vezhnevets et al., 2017). In HDRL, goals are divided into high-level and low-level sub-goals, each handled by different models, as shown in Figure 2.2. The high-level model inputs the state of the environment and outputs high-level actions or sub-goals based on its policy. These decisions are passed down to a low-level model, which executes the specific actions needed to achieve the sub-goals (Kulkarni et al., 2016; Ma et al., 2024b). For instance, a robot's high-level goal could be "make a sandwich," with sub-goals like "get bread," "get cheese," and "get ham," and each sub-goal having specific sub-policies such as "move to the fridge" $\rightarrow$ "open the fridge" $\rightarrow$ "take out cheese," all executed by the low-level model. The HDRL process has several steps: Initially, policies and value functions for both high-level and low-level models are trained using DRL algorithms like Proximal Policy Optimization (PPO) or Deep Q-learning (DQN). The high-level model observes the global state and selects sub-goals to maximize overall reward. These sub-goals are communicated to the low-level model, which perform the specific actions. The low-level model interacts with the environment, observing states and rewards. This reward is shared between high-level and low-level actions, providing feedback for both models. Policies are updated based on observed rewards and state transitions to enhance performance over time. An example of HDRL is the Hierarchical DQN (h-DQN) (Kulkarni et al., 2016; Zhao et al., 2023a), which combines hierarchical decision-making with the Deep Q-learning algorithm. This approach is effective in environments with sparse and delayed rewards, aligning short-term objectives with long-term goals for more efficient learning

and better performance. Inspired by HDRL concepts, a two-level HDRL framework is adopted to enhance performance. The framework integrates methods like Oracle Guiding, Adaptive Deep Monte Carlo (ADMC), Relative Advantage Reward Shaping (RARS), and PPO to improve the performance in complex imperfect-information card games such as DouDiZhu and Big2. ADMC and PPO are distinct Reinforcement Learning methods that contribute to the hierarchical structure of HDRL.

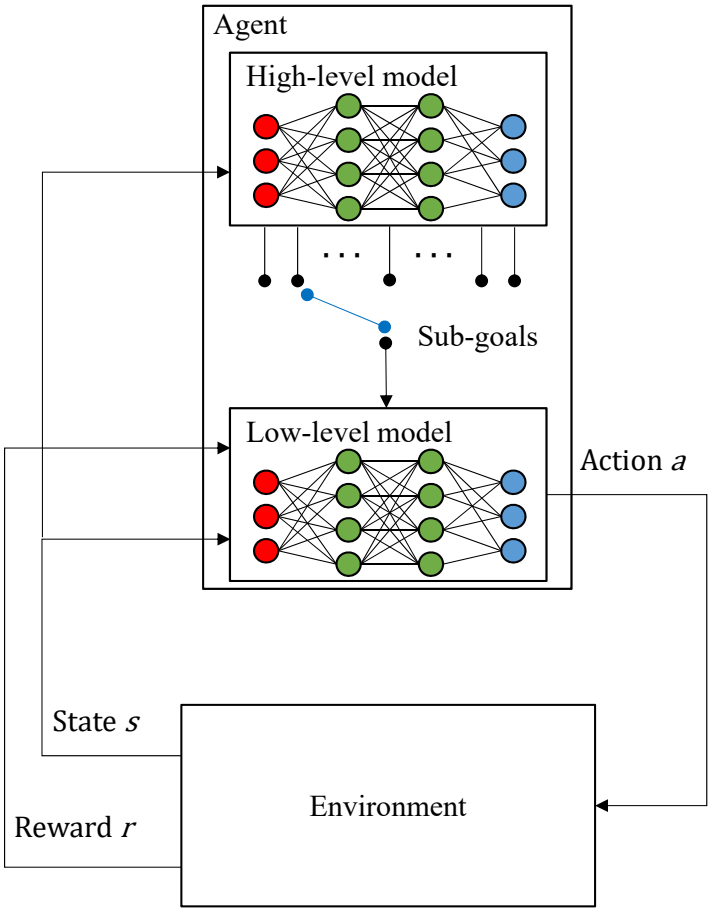

Figure 2.2: Hierarchical Deep Reinforcement Learning. Hierarchical Deep Reinforcement Learning (HDRL) divides decision-making into a high-level model that sets sub-goals and a low-level model that executes specific actions to achieve these sub-goals.