Game Reinforcement Learning: Modelling, Optimizing, Proving and Solving

Abstract

Deep reinforcement learning has achieved remarkable outcomes in single agent games like Atari and GO. However, in real world, multi-agent scenarios are more common. Recently, the excellent performance of multi-agent reinforcement learning in real-time strategy game, online advertising and text generation shows it great potential in artificial general intelligence. Due to the conflicting goals between agents and actions that affect each other, researchers look into game theory for inspiration. Concerning the combination of multi-agent reinforcement learning and game theory, there are mainly four questions: 1) how to build a unified model; 2) how to optimize under the unified model; 3) how to prove the existence of solution; 4) how to solve in polynomial time. For each question, I put forward some specific solutions: model-based reinforcement learning and stochastic game for question 1; correlated equilibrium, trembling hand perfect, pareto optimal, inverse reinforcement and "tit for tit" or "grim trigger" strategy for question 2; Kakutani's fixed point theorem and Lyapunov functions for question3; convex and non-convex optimization for question 4.

Keywords: multi-agent reinforcement learning; game theory; Kakutani's fixed point theorem; Lyapunov functions; convex and non-convex optimization

1 Introduction

In single agent reinforcement learning, an agent interacts with environment and maximizes the cumulative returns according to optimal action policy. While in multi-agent reinforcement learning, a set of autonomous agents share a common environment and respectively maximize their own cumulative returns, which may not be achievable. Even though multi-agent reinforcement learning has been successfully applied in many fields, multi-agent reinforcement learning is fundamentally difficult since agents not only interact with the environment but also with each other. For instance, if single agent Q learning is directly deployed by considering other agents as a part of the environment, it breaks the theoretical convergence guarantees and makes the learning unstable, because the changes in strategy of one agent would affect the strategies of other agents and vice versa. And sometimes, it may even lead to "theatre effect".

The introduction of game theory provides an effective approach to analyze the interactions among agents. Considering the ingredients of game theory have corresponding relationships with reinforcement learning: environment and information, actions and policy, agents and players, rewards and pat-off, many models and optimization methods in game theory can transfer to multi-agent reinforcement learning easily. Especially the idea of equilibrium solutions. Comparing to optimal solutions, equilibrium solutions are more reasonable in finding an effective strategy.

Dynamic programming and gradient descent are common methods to solve the equilibrium of game reinforcement learning. However, before solving, it's a must to prove the existence of the solution(equilibrium). Considering the proof of Nash equilibrium, we find fixed point theorem counts. Besides, in matrix game, the Lyapunov functions and Lyapunov stability theorems are vital. When it comes to solving, quadratic programming used to work well. However, when dimension

explodes, the time complexity is quite large. Gradient descent might be a more efficient way, while it still could fail when the feasible zone is non-convex.

In short, building a game reinforcement learning model and make optimization based on the unified model are only the first two-steps. How to prove the existence of equilibrium, together with how to solve the equilibrium in polynomial time still have long way to go.

2 Research aims and questions

2.1 Aims

Based on my former study, there are still a lot of gaps in the combination of game theory and multi-agent reinforcement learning, and I am committed to filling those gaps in the following aspects.

First, build a unified model based on Stochastic Game to elaborate why the two subjects can be connected and how to make combination and I have been writing a review on game reinforcement learning.

Second, refine the solution concepts and optimize the pay-off (reward) function, strategy(action), etc. For example, subgame perfect, trembling hand perfect and pareto optimal are beneficial in the refinement of equilibrium; value function decomposition reduces the computational complexity of joint value functions and inverse reinforcement learning compensates the limitation of artificially designed reward function; “tit for tit” and “grim trigger” strategy may be more effective in the exploitation and exploration issues compared to greedy strategy.

Third, study the existence and computability of equilibrium in game reinforcement learning. In game theory, without violating the assumption that people are rational, equilibrium seems to be the best solution. Although some pioneers did make contributions such as Nash Q learning, the constraints were very harsh and in the era of dimension explosion, the time complexity is unbearable. As far as I know, fixe point theorem and Lyapunov functions can be applied to prove existence. Convex and non-convex optimization can be applied in gradient descent. Although I’m not good at theoretical analysis, I have been studying real variable function, functional analysis, matrix decomposition and optimization theory to prepare in advance to solve these problems.

I have more ideas than I can list. But one thing must be stressed is to think independently. I will not blindly follow the heated trends, nor merely pursue the number of papers as before. I do take academic research seriously and consider academic fruits are natural, not utilitarian. My goal is simply to help fill in the gaps in game reinforcement learning and strengthen my ability in quickly acquiring new knowledge and keeping long term learning.

2.2 Questions

1 How to combine game theory with multi-agent reinforcement learning in a unified model?

Key: Model-based Reinforcement Learning; Stochastic Game

2 How to refine solution concepts, optimize pay-off (reward) function and strategy(action)?

Key: Correlated equilibrium; Subgame Perfect, Trembling Hand Perfect and Pareto Optimal; Inverse Reinforcement; “Tit for Tit” and “Grim Trigger” Strategy

3 How to prove the existence of equilibrium?

Key: Kakutani’s Fixed Point Theorem; Lyapunov Functions

4 How to solve the equilibrium in polynomial time?

Key: Convex and Non-convex Optimization

3 Literature Review

According to *Game Theory* [1]and *A Course in Game Theory* [2], some typical game models and equilibrium concepts are summarized in table 1, which also clearly shows how Stochastic Game develops. In general, there are two types of games: Strategic Game and Extensive Form Game. The former is considered as static game and actions are conducted by each player simultaneously, while the latter is considered as dynamic game and actions are conducted by each player sequentially. The information type is either perfect or imperfect, which is similar to Markov decision process (MDP) and partially observed Markov decision process (POMDP) in reinforcement leaning. Nash equilibrium is a basic solution concept with many improvements, different games have different equilibrium solutions. There are also two types of strategy: pure strategy and mixed strategy. In game reinforcement learning, Stochastic Game [3] is of great importance.

<table>
<tr><td>Game Type</td><td colspan="2">Strategic Game:
static, simultaneous</td><td colspan="3">Extensive Form Game:
dynamic, sequential</td></tr>
<tr><td>Information Type</td><td>Perfect</td><td>Imperfect</td><td colspan="2">Perfect</td><td>Imperfect</td></tr>
<tr><td>Solutions</td><td colspan="5">Best response, Nash Equilibrium</td></tr>
<tr><td>Strategy</td><td colspan="5">Pure Strategy, Mixed Strategy</td></tr>
<tr><td>Equilibriums</td><td>Nash Equilibrium</td><td>Bayesian Equilibrium</td><td colspan="2">Subgame Perfect Equilibrium</td><td>Perfect Bayesian Equilibrium</td></tr>
<tr><td>Typical Games</td><td>Strictly Competitive Game</td><td>Bayesian Game</td><td colspan="2">Repeated Game</td><td>Bayesian extensive Game with observable actions</td></tr>
<tr><td rowspan="3">Typical Equilibriums</td><td colspan="2">Mixed Strategy Nash Equilibrium
Correlated Equilibrium
Evolutionary Equilibrium</td><td colspan="3">Sequential Equilibrium</td></tr>
<tr><td colspan="5">Trembling Hand Perfect; Pareto optimal</td></tr>
<tr><td colspan="2"></td><td colspan="2">Markov Perfect Equilibrium</td><td></td></tr>
<tr><td>Time Type</td><td colspan="2"></td><td colspan="2">Discrete</td><td>Continuous</td></tr>
<tr><td>Markov Games</td><td colspan="2"></td><td>Stochastic Game</td><td>Sequential Game with separable pay-off</td><td>Differential Game</td></tr>
</table>

Table 1 Models and Equilibriums in Game Theory

Based on *Reinforcement Learning: An introduction* [4]and *Reinforcement Learning: State-Of-The-Art* [5], the basic framework and its characteristics are concluded in table 2.

Basic Framework	Characteristics
MDP	Single agent, multiple states, perfect information
POMDP	Single agent, multiple states, imperfect information
Repeated Game	Multi agent, one state, perfect information
Stochastic Game	Multi agent multiple states, perfect information

Table 2 Sequential Decision Making

3.1 Stochastic Potential Game (SPG)

In a game, if each player changes his or her own goals or strategies, it can be mapped to a global function, which is called potential function, and this game is called potential game. In general, potential games can be regarded as the "single agent component" of multi-agent games [6], because the interests of all agents in SPG are described as a single potential function. When potential games are extended to stochastic potential games, the complexity of the problem increases.

Macua et al. [7] studied the extensive form of potential game and proved the existence of Nash equilibrium under this premise, and theoretically proved that Nash equilibrium in pure strategy potential game can be found by solving MDP [8]. Mazumdar et al. [9] proposed a strategy-based dynamic update algorithm for potential games and applied it in Morse-Smale games, proving that the algorithm can converge to local Nash equilibrium. Chen et al. [10] proposed centralized training and exploration, and conducted decentralized actions by policy distillation to promote coordination and effective learning among agents.

3.2 Extensive Game with Imperfect Information

There are CFR series and FSP series algorithms to solve imperfect information games, and the verification environment of the two series algorithms are mainly Texas Holdem.

3.2.1 Counterfactual Regret (CFR)

CFR [11] algorithm combines regret minimization algorithm, minimizes the global regret value by minimizing the regret value on a single information set, and finally makes the average strategy in the game close to Nash equilibrium. At the need of traversing the whole game tree, large time complexity and slow convergence are main disadvantages.

Algorithms	Advantages	Disadvantages
CFR	The algorithm combines regret minimization algorithm and average strategy	Prior knowledge and perfect recall are required; Have to traverse the entire game tree
Lazy-CFR [12]	With lazy update, it doesn't have to traverse the entire game tree	Prior knowledge and perfect recall are required
MCCFR [13]	Monte Carlo sampling is used to reduce the time complexity of R algorithm	Prior knowledge and perfect recall are required; big variance
VR-MCCFR [14]	The problem of high variance is alleviated by taking the average utility value of no visiting nodes as the baseline.	Prior knowledge and perfect recall are required
DNCRM [15]	Based on double neural network, no need a lot of prior knowledge, and the convergence speed is faster	Perfect recall is required
Logistic-CFR [16]	Use regression trees as function approximators	Prior knowledge is required; Have to traverse the entire

		game tree
Deep-CFR [17]	Using neural networks as function approximators	Perfect recall is required
SD-CFR [18]	The average strategy is extracted from the iterative Q-value network buffer, which has low error and improves the convergence speed of deep-CFR	Imperfect information game cannot be handled
DREAM [19]	Converges to Nash equilibrium in imperfect information with low variance	Have to traverse the entire game tree
LONR [20]	Converge without perfect recall	Update rules similar to Q learning; convergence time needs to be improved

Table 3 Advantages and disadvantages of CFR series algorithms

3.2.2 Fictitious Self-Play (FSP)

Fictitious Play (FP) [21] is an algorithm that performs optimal responses to the adversary's average strategies to solve the Nash equilibrium. After repeated iterations, the algorithm's average strategies in two-person zero-sum games and potential games will converge to the Nash equilibrium.

Algorithms	**Advantages**	**Disadvantages**
FP	Nash equilibrium is solved by the optimal response to the opponent's average strategy	High dimensional problems cannot be applied to regular representations that rely on real scenarios
EFP [13]	The concept of virtual chess is extended to extensive game	States are represented in lookup tables, and average strategy updates traverse the entire game tree
FSP [22]	Reinforcement learning and supervised learning are used to replace optimal response calculation and average strategy updating respectively	Player and the opponent are required to follow an order of action, so it is not suitable for games with imperfect information
NFSP[17]	Approximate solution with neural network	The optimal reaction depends on the calculation of deep Q learning; the convergence time is long
MC-NFSP [23]	Combined with NFSP and Monte Carlo tree search, convergence can be achieved in Othello chess where NFSP cannot converge	Large variance of Monte Carlo search cannot be overcome
ANFSP [23]	Combined with NFSP and Monte Carlo tree search, convergence can be achieved in Othello chess where NFSP cannot converge	Convergence rate can be further optimized
LOLA [24]	Using modeling ideas, consider the learning processes of other agents	When an agent updates its own strategy, it takes a long time to make a decision considering the learning process of other agents

Table 4 Advantages and disadvantages of FSP series algorithms

Although the imperfect information extensive game represented by Texas Hold'em poker has made breakthrough progress under CFR and NFSP series algorithms, it has not completely solved, and still the difficulties and key points. In order to solve this kind of game, two problems may need to be solved: first, how to quantify the uncertainty under imperfect information and the non-stationarity of environment; second, how to ensure the communication and collaboration between agents efficiently.

3.3 Value Decomposition Methods

The value function decomposition methods have great advantages in multi-agent reinforcement learning in cooperative environment. It can solve the problems such as partial observable environment, action space exponential explosion, instabilities algorithm and credit assignment in multi-agent reinforcement learning. Therefore, in recent years, researchers are devoted to the study value function decomposition methods. Combined with other mechanisms, many valuable algorithms are proposed.

Algorithms	Advantages	Disadvantages
VDN [25]	The joint value function is the linear sum of each agent value function	Low efficiency, few games that meet the prerequisites
QMIX[26]	Neural network is applied to approximate the joint value function, the efficiency gets improved	Joint valued functions are required to be monotone to individual valued functions under strict conditions
QTRAN[27]	The VDN method is used to obtain the combined value function of the sum, and then the neural network is used to fit the difference between the sum of combined value function and the combined value function, which has the respective advantages of VDN and QMIX	Fail to overcome the shortcomings of VDN and QMIX, convergence conditions are too harsh
Qatten [28]	A hybrid value function network based on multi-attentional mechanism is proposed to approximate the joint value function and decompose the single value function, and the generalized form of the joint value function and the value function of any number of agents is derived theoretically for the first time	No exploration mechanism to make the algorithm perform better in complex tasks

Table 5 Advantages and disadvantages of VDN series algorithms

3.4 Experimental platforms

There are many experimental platforms for multi-agent reinforcement learning. I conclude 17 platform in total. And I have made experiments on 5 of them: Gridworld, Particle MPE, MAgent, StarCraft II and Gym. I hope I can test on more platforms in the future.

Platforms	Descriptions
Grid World	The status information is mainly agent coordinates, and the action can be four-way or eight-way.
Multi-agent Reinforcement	Dozens of small Python-based Grid World environments available.

Learning	
DeepMind MAS	Multi-agent environment used in paper [25].
Particle MPE	Pellet environment, used in MADDPG, is a more complex Grid World environment.[29]
MAgent	Mainly study competition and collaboration when the environment is composed of a large number of agents, used in paper [30].
Pommerman	The bomber environment is a competition environment for NIPS 2018. The environment is mainly 2v2, with partial observable setting and also have a communication scene.
Multiagent emergence	The hide-and-seek environment of OpenAI [31].
Quake III Arena Capture the Flag	From DeepMind's Lab environment, one of the maps is of the Quake III Arena. 2v2, agents compete to capture the flag in the first-person view in two indoor and outdoor scenes[32][33].
Google Research Football	Modified and encapsulated from an earlier football mini-game, it can be mainly divided into 11v11 single agent scenario and 5v5 multi-agent scenario[34].
Neural MMOs	OpenAI open source a large complex multi-agent game scene.
StarCraft II	A representative environment for real-time strategy, Alpha Star [35] has made a remarkable performance. There are also many well-known algorithms based on this environment, such as QMIX[26] and COMA.
Multi-agent Combat Arena	Heterogeneous multi-agent distributed decision and control technology reintegration platform.
Unity ML-Agents Toolkit	It's not just an environment, but a game engine, an IDE for making games. There are a lot of games out there are based on Unity, especially mobile games.
Fever Basketball	From NetEase Fuxi Laboratory. There are not only a variety of roles and positions (PG, SG, C, PF, SF) to choose from, but also a variety of scenarios (1v1, 2v2, 3v3) for training.
Botzone	The competition environment of 2020 IJCAI, opened by Artificial Intelligence Laboratory of Peking University, with more than 20 games.
OpenAI Gym	A kit for research and development and intensive learning algorithms. It trains agents to do everything from playing Pong or go.
Petting Zoo	An integration of multi-agent environments, including MAgent.

Table 6 Experimental platforms

There have many platforms to test on and each setting or environment is quite different. How to propose a more general model or algorithm still need more efforts.

4 Project design

4.1 How to combine Game Theory with Reinforcement Learning?

Currently, many reinforcement learning algorithms have no specific environment and regard datasets as environment, which is known as model-free reinforcement learning. Such methods, combined with neutral network, are more like supervised learning instead of reinforcement learning.

Because of big data, the results of model-free methods are just statistical patterns.

To build a unified game reinforcement learning model, model-based reinforcement learning has advantages in stable environment. And Stochastic Game is essential foundation too.

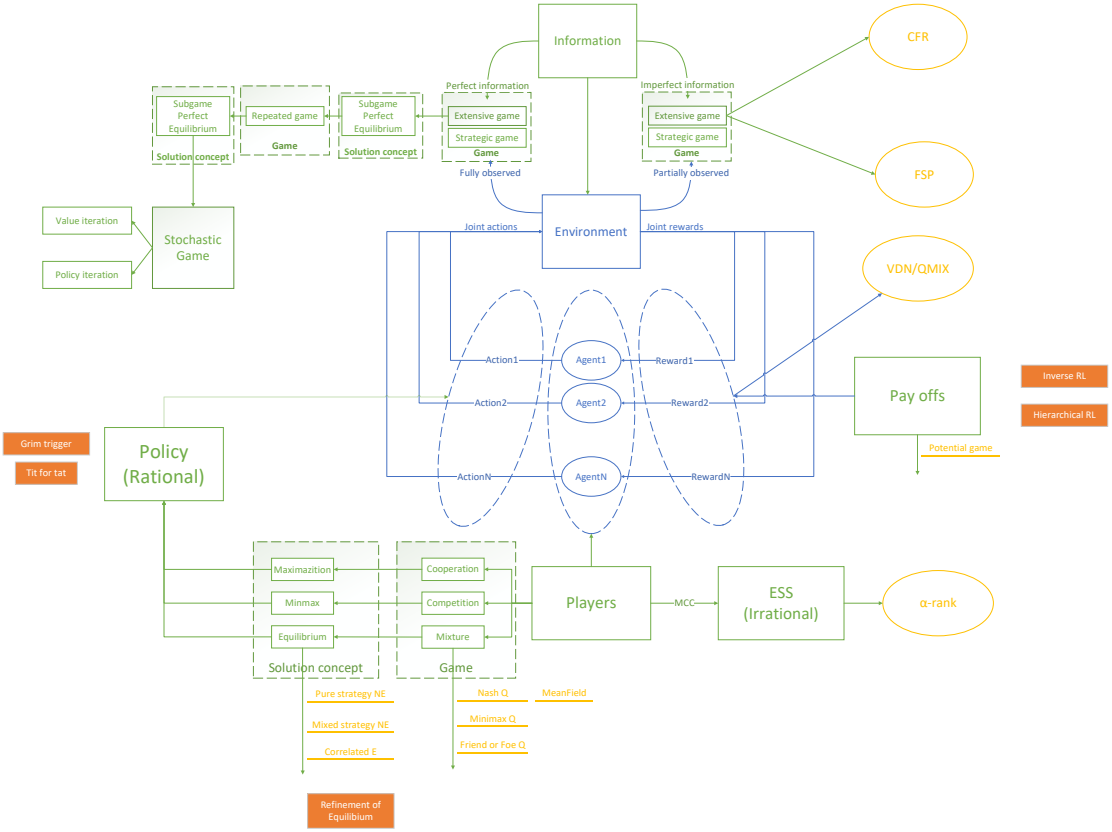

Figure 1 The unified model of Game Reinforcement Learning

According to figure 1, Blue part is the framework of multi-agent reinforcement. Green part is the framework of game theory. Yellow part is the typical algorithms of game reinforcement learning. And the orange part is the potential direction that I can make improvement.

Multi-agent reinforcement learning and game theory seem to have a natural connection: environment and information, actions and policy, agents and players, rewards and pat-offs have one to one correspondence respectively. Whether reinforcement learning or game theory has plenty of methods to analyze and optimize questions, so there is no doubt the combination inspires us in many aspects.

4.2 How to refine solution concepts, optimize pay-off (reward) function and strategy(action)?

Many methods in game theory can transfer to reinforcement learning easily due to similar framework. Since game theory acts as "foreign aid" for multi-agent reinforcement learning, we can find assists from it definitely. For example, in a Nash equilibrium, each player chooses a strategy independently. However, this approach sometimes does not yield very good returns, and many bad strategies are chosen. In fact, players can "make a pact" not to choose a bad outcome, which will increase revenue. Then correlated equilibrium is proposed, and how to solve correlated equilibrium worth studying. Similarly, the optimization in joint reward function and in the issue of exploitation and exploration also worth further study.

According to figure 1, the combination of game theory and multi-agent reinforcement learning

does have achieved rich academic results. And it is worth noting that these algorithms come in families, just like DQN before. So, I will try to integrate all improvements into one framework, just as Rainbow-DQN before.

4.3 How to prove the existence of equilibrium?

Without knowing the existence of an equilibrium, it is difficult (perhaps meaningless) to understand its properties. Armed with fixed point theorem, we know that every finite game has a mixed strategy Nash equilibrium, and thus we can simply try to locate it. However, Nash equilibrium is only a basic solution concepts under certain conditions, what if the conditions change or other equilibrium concepts like correlated equilibrium more suitable for certain problems? We need more theories to prove the existence of equilibrium. In matrix game, Lyapunov functions is useful, which may give us some inspiration.

Kakutani's Fixed Point Theorem are used to prove the existence of Nash equilibrium in finite (strategic form) games, and the definition is as follows:

Kakutani's Fixed Point Theorem: Define a best response correspondence $B: \Sigma \rightrightarrows \Sigma$, the sufficient conditions for a best response correspondence B to have a fixed point are:

(1) Σ is a compact, convex, nonempty subset of a (finite dimensional) Euclidean space.

(2) $B(\,)$ is nonempty for all σ。

(3) $B(\sigma)$ is convex for all σ.

(4) $B(\sigma)$ has a closed graph.

For condition (1), a set in a Euclidean space is compact if and only if it is bounded and closed. A set Σ is convex if for any $x, y \in \Sigma$ and any $\lambda \in [0,1]$, $\lambda x + (1 - \lambda)y \in \Sigma$.

For condition (2), every player has a best response to the other players' strategies, whatever those strategies are.

For condition (3), $B(\sigma)$ is convex-valued correspondence (aka $B(\sigma)$ is convex set) for all σ. And equivalently, $B(\sigma) \subset \Sigma$ is convex if and only if $B_i(\sigma_{-i})$ is convex for all i.

For condition (4), $B(\sigma)$ has a closed graph: that is, if $\{x, y\} \rightarrow \{x^*, y^*\}$ with $y \in B(x)$, then $y^* \in B(x^*)$.

In general scenarios, the conditions can be changed, as either Σ or $B(\sigma)$ can be non-convex. How to prove the existence of Nash equilibrium in more general conditions worth studying.

Convergence Analysis via Lyapunov Functions in Matrix Game

In general, when the matrix is nonsingular, there are 4 different types of equilibrium points:

#	**Equilibrium Point**	**Eigenvalues λ_1, λ_2**
1	Node	λ_1, λ_2 are real numbers of the same sign ($\lambda_1 \cdot \lambda_2 > 0$)
2	Saddle	λ_1, λ_2 are real numbers of the non-zero of opposite sign ($\lambda_1 \cdot \lambda_2 < 0$)
3	Focus	λ_1, λ_2 are complex numbers, the real parts are equal and non-zero ($Re\lambda_1 = Re\lambda_2 \neq 0$)
4	Center	λ_1, λ_2 are purely imaginary numbers, ($Re\lambda_1 = Re\lambda_2 = 0$)

Table 7 different types of equilibrium points

The Phase Portraits of Equilibrium Points is shown in figure 2.

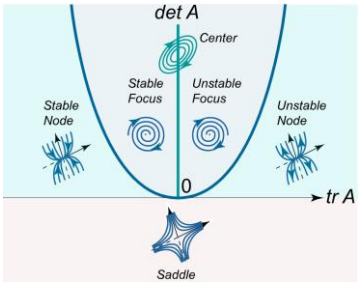

Figure 2 Phase Portraits of Equilibrium Points

In the case of purely imaginary roots (when the equilibrium point is a center), we are dealing with the classical stability in the sense of Lyapunov. Lyapunov Functions and Lyapunov Stability Theorems are no further elaboration here, when it comes to matrix games, I will study deeper.

4.4 How to solve the equilibrium in polynomial time?

While Nash Q learning addressed two-player general-sum games, it still had theoretical limitation on single equilibrium only and applied quadratic programming to solve general-sum equilibrium. As the dimensions explode, the computation time of quadratic programming is intolerable.

In machine learning, gradient descent is widely used in finding the optimal solution. Based on convex optimization, it follows a very important theorem that any local optimal solution is a global optimal solution. However, in game reinforcement learning settings, the feasible zone can be non-convex with an infinite number of local optima, so that the time complexity for solving global optima is exponential (NP hard). In order to solve the equilibrium in polynomial time, there needs more effective methods.

5 Timeline

Table 8 shows time and task schedule.

Time	Task
2022	1.Research on question 1 and question 2. 2.Write reviews. 3.Learn game theory and multi-agent reinforcement learning 4.Learn matrix factorization, fixed point theorem and convex optimization. 5.Get familiar with experimental platform.
2023	1.Research on question 2 or other questions. 2. Learn game theory and multi-agent reinforcement learning 3.Make theoretical proof and experimental verification based on 3.4.
2024	1.Research on question 3 and question 4 or other questions. 2.Learn fixed point theorem, Lyapunov Functions, convex optimization and so forth. 3.Make theoretical proof and experimental verification based on 3.4.
2025	1.Based on former study, write graduation thesis 2.Work as a teaching assistant. 3.Get an internship at an Internet company.

Table 8 Timeline

6 Expected outcomes/impact

The expected outcomes are listed in three main aspects in table 9.

Publications	Top conference papers and SCI papers
Competitions	Participate in Kaggle or Top conference competitions and win award.
Projects and experiences	Work as a teaching assistant in universities. Get internships in Internet company.

Table 9 Expected outcomes

References

[1] Drew Fudenberg, Jean Tirole. Game theory[M]. MIT Press, 1991.

[2] Martin J. Osborne, Ariel Rubinstein. A Course in Game Theory[M]. MIT Press, 1994.

[3] Shapley, Lloyd S. Stochastic games[C]. Proceedings of the national academy of sciences 39(10): 1095-1100, 1953.

[4] Richard Sutton, Andrew G. Barto. Reinforcement Learning: An Introduction[M]. MIT Press, 1998.

[5] Marco A. Wiering, Martijn van Otterlo. Reinforcement Learning: State-Of-The-Art[M]. Springer-Verlag Press, 2012.

[6] Ozan Candogan, Ishai Menache, Asuman E. Ozdaglar, et al. Flows and decompositions of games:harmonic and potential games[J]. Mathematics of Operations Research, 36(3): 474-503, 2011.

[7] Sergio Valcarcel Macua, Javier Zazo, Santiago Zazo. Learning parametric closed-loop policies for Markov potential games[C]. The 6th International Conference on Learning Representations, 2018.

[8] David S. Leslie, Edmund J. Collins. Generalised weakened fictitious play[J]. Games and Economic Behavior, 56(2):285-298, 2016.

[9] Eric Mazumdar, Lillian J. Ratliff, S. Shankar Sastry. On Gradient-Based Learning in Continuous Games[J]. SIAM Journal on Mathematics of Data Science, 2(1): 103-131, 2020.

[10] Gang Chen. A New Framework for Multi-Agent Reinforcement Learning-Centralized Training and Exploration with Decentralized Execution via Policy Distillation[C]. The 19th International Joint Conference on Autonomous Agents & Multiagent Systems, 1801-1803, 2020.

[11] Martin Zinkevich, Michael Johanson, Michael H. Bowling, et al. Regret Minimization in Games with Incomplete Information[C]. The 21st Annual Conference on Neural Information Processing Systems, 1729-1736, 2007.

[12] Yichi Zhou, Tongzheng Ren, Jialian Li, et al. Lazy-CFR: fast and near-optimal regret minimization for extensive games with imperfect information[C]. The 8th International Conference on Learning Representations, 2020.

[13] Johannes Heinrich, Marc Lanctot, David Silver. Fictitious Self-Play in Extensive-Form Games[C]. Proceedings of the 32nd International Conference on Machine Learning, 805-813, 2015.

[14] Martin Schmid, Neil Burch, Marc Lanctot, et al. Variance reduction in Monte Carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines[C] The Thirty-Third AAAI Conference on Artificial Intelligence, 2157-2164, 2019.

[15] Hui Li, Kailiang Hu, Shaohua Zhang, et al. Double Neural Counterfactual Regret Minimization[C]. The 8th International Conference on Learning Representations, 2020.

[16] Marc Lanctot, Kevin Waugh, Martin Zinkevich, et al. Monte carlo sampling for regret

minimization in extensive games[C] The 23rd Annual Conference on Neural Information Processing Systems, 1078-1086, 2009.
[17] Johannes Heinrich, David Silver. Deep reinforcement learning from self-play in imperfect-information games[J]. CoRR: 1603.01121, 2016.
[18] Eric Steinberger. Single deep counterfactual regret minimization[J]. CoRR: 1901.07621, 2019.
[19] Eric Steinberger, Adam Lerer, Noam Brown. DREAM: Deep Regret minimization with Advantage baselines and Model-free learning[J]. CoRR: 2006.10410, 2020.
[20] Ian A. Kash, Michael Sullins, Katja Hofmann. Combining No-regret and Q-learning[C]. The 19th International Joint Conference on Autonomous Agents & Multiagent Systems, 593-601, 2020.
[21] Ulrich Berger. Brown's original fictitious play[J]. Game theory and information, 135(1):572-578, 2005.
[22] Drew Fudenberg, David K. Levine. Consistency and cautious fictitious play[J]. Journal of Economic Dynamics and Control, 19(5/7), 1065-1089, 1996.
[23] Li Zhang, Yuxuan Chen, Wei Wang, et al. A Monte Carlo Neural Fictitious Self-Play approach to approximate Nash Equilibrium in imperfect-information dynamic games[J]. Frontiers of Computer Science, 15(5): 155334, 2021.
[24] Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, et al. Learning with Opponent-Learning Awareness[C]. The 17th International Joint Conference on Autonomous Agents & Multiagent Systems, 122-130, 2018.
[25] Peter Sunehag, Guy Lever, Audrunas Gruslys, et al. Value-Decomposition Networks for Cooperative Multi- agent Learning Based on Team Reward[C]. The 17th International Joint Conference on Autonomous Agents & Multiagent Systems, 2085-2087, 2018.
[26] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, et al. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning[C]. Proceedings of the 35th International Conference on Machine Learning, 4295-4304, 2018.
[27] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, et al. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning[C]. Proceedings of the 36th International Conference on Machine Learning, 5887-5896, 2019.
[28] Yaodong Yang, Jianye Hao, Ben Liao, et al. Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning[J]. CoRR abs/2002.03939, 2020.
[29] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments[C]. The 31st Annual Conference on Neural Information Processing Systems, 6379-6390, 2017.
[30] Yaodong Yang, Rui Luo, Minne Li, et al. Mean Field Multi-Agent Reinforcement Learning[C]. Proceedings of the 35th International Conference on Machine Learning, 5567-5576, 2018.
[31] Bowen Baker, Ingmar Kanitscheider, Todor M. Markov, et al. Emergent Tool Use from Multi-Agent Autocurricula[C]. The 8th International Conference on Learning Representations, 2020.
[32] Charles Beattie, Joel Z. Leibo and Denis Teplyashin, et al. DeepMind Lab. 2016.
[33] Jaderberg M, Czarnecki W M, Dunning I, et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning[J]. Science, 364(6443):859-865, 2019.
[34] Hangyu Mao, Wulong Liu, Jianye Hao, et al. Neighborhood Cognition Consistent Multi-Agent Reinforcement Learning[C]. The Thirty-Fourth AAAI Conference on Artificial Intelligence, 7219-7226, 2020.

[35] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature. 575(7782): 350-354, 2019.