Multi-Agent Reinforcement Learning
CS5730 Paper

Sherman Kettner[1]

University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO 80918 USA
skettner@uccs.edu

Introduction

This research paper explores the topic of **Multi-Agent Reinforcement Learning** (MARL), a dynamic and increasingly influential subfield of reinforcement learning (RL). MARL is particularly relevant to game theory, where multiple decision-makers interact in complex environments. The topic was selected both due to its alignment with the author's background in artificial intelligence (AI) techniques and its applicability to ongoing research and practical projects.

At its core, reinforcement learning models decision-making as a discrete-time process, in which an agent interacts with an environment by selecting actions that yield scalar rewards (Tan 1993). The agent's objective is to maximize cumulative reward through trial-and-error learning. MARL extends this paradigm to settings involving multiple agents, each of which learns and adapts within a shared environment (Yang and Wang 2020). These agents may be cooperative, competitive, or engage in interactions that combine both dynamics.

MARL is closely linked to game theory, which studies how rational agents make decisions in strategic settings. In MARL, each agent's reward typically depends not only on its own actions but also on the actions of others, creating interdependencies typical of game-theoretic environments (Yang and Wang 2020). Concepts such as Nash equilibria, Pareto efficiency, and best-response strategies often emerge in the analysis of multi-agent systems, particularly in competitive or mixed settings (Lanctot et al. 2017; Tuyls and Parsons 2007). As a result, MARL provides a computational framework for studying learning and adaptation in repeated games and dynamic interactions (Hu and Wellman 1998).

Applications of MARL are rapidly expanding across various domains, including user-centric radio access technology (Caso et al. 2021), unmanned aerial vehicles, strategic games such as Go and poker (Zhang, Yang, and Başar 2021), autonomous driving (Yang and Wang 2020), algorithmic trading, distributed control systems (Buşoniu, Babuška, and De Schutter 2010), multi-robot coordination, and cooperative communication networks (Sven and Klaus 2022). Its relevance to real-world problems stems from the need to model systems in which independent decision-makers must reason about one another's behavior in dynamic, uncertain environments.

A central insight in MARL research is that agent interactions fall into three primary categories:

- **Cooperative** – agents share a common objective and must learn to coordinate their actions.
- **Competitive** – agents act in opposition, often in adversarial or zero-sum scenarios.
- **Mixed** – agents have partially aligned goals, requiring both collaboration and strategic self-interest.

Despite its promise, MARL introduces unique research challenges that complicate learning and convergence:

- **Non-stationarity** – the environment becomes non-stationary as each agent's policy evolves, violating standard RL assumptions.
- **Learning Communication** – agents may benefit from developing communication protocols to share local observations or intentions.
- **Coordination** – achieving coordinated behavior in cooperative and mixed-motive settings is non-trivial, particularly with decentralized learning.
- **Credit Assignment** – it is often difficult to determine the contribution of each agent's action to a shared outcome or reward.
- **Scalability** – the computational and strategic complexity increases sharply with the number of agents.
- **Partial Observability** – agents typically operate with limited or noisy information about the environment and other agents' internal states (Sven and Klaus 2022).

MARL's integration of reinforcement learning with game-theoretic reasoning makes it a powerful framework for modeling intelligent, adaptive systems. As such, it represents an important frontier in both AI research and practical multi-agent applications.

The sections that follow present a formal model for MARL, a historical overview of its development, analysis of key results, and a discussion of future directions.

Model

To analyze learning and strategic behavior in multi-agent environments, we require a formal model that captures both environmental dynamics and inter-agent interactions. The **Stochastic Game**—also known as a **Markov Game**—serves as a unifying structure for this purpose. It generalizes the Markov Decision Process (MDP) to multiple agents, allowing us to rigorously define policies, value functions, and equilibrium concepts in a setting that merges reinforcement learning with game theory (Alonso and Njupoun 2024; Sven and Klaus 2022; Caso et al. 2021; Hu and Wellman 1998; Zhang, Yang, and Başar 2021; Buşoniu, Babuška, and De Schutter 2010; Yang and Wang 2020; Busoniu, Babuska, and De Schutter 2006).

While MDPs are used for single-agent reinforcement learning, Stochastic Games extend the framework to allow multiple agents to interact within a shared environment. Each agent has its own reward function and strategy, and may be cooperative, competitive, or mixed in relation to other agents. Many papers highlight this model's centrality in formalizing strategic behavior and learning in MARL (Zhang, Yang, and Başar 2021; Buşoniu, Babuška, and De Schutter 2010; Yang and Wang 2020).

Markov Games: Formal Definition

A Markov Game is defined by a 5-tuple:

$$(S, A, P, R, \gamma)$$

Where:

- S is the finite set of environment states shared by all agents.
- $A = A_1 \times \ldots \times A_n$ is the joint action space, where A_i is the finite set of actions available to agent i.
- $P : S \times A \to \Delta(S)$ is the transition probability function, where $P(s' \mid s, \vec{a})$ gives the probability of moving to state s' given current state s and joint action $\vec{a}$.
- $R = (R_1, \ldots, R_n)$ is the vector of reward functions, with $R_i : S \times A \to R$ defining the immediate reward for agent i.
- $\gamma \in [0, 1)$ is the discount factor, representing how much future rewards are weighted relative to immediate ones.

Each agent i follows a policy $\pi_i : S \to \Delta(A_i)$, which maps states to probability distributions over actions. The joint policy is $\vec{\pi} = (\pi_1, \ldots, \pi_n)$.

To evaluate expected outcomes, we define the *expected discounted return* for agent i as:

$$G_i = \sum_{k=0}^{\infty} \gamma^k R_{i,t+k+1}$$

The state-value function for agent i under policy $\vec{\pi}$ is:

$$V_i^{\vec{\pi}}(s) = E\left[\sum_{k=0}^{\infty} \gamma^k R_{i,t+k+1} \,\middle|\, S_t = s, A_{t:\infty} \sim \vec{\pi}\right]$$

The action-value function (Q-function) is:

$$Q_i^{\vec{\pi}}(s, \vec{a}) = E\left[\sum_{k=0}^{\infty} \gamma^k R_{i,t+k+1} \,\middle|\, S_t = s, A_t = \vec{a}, A_{t+1:\infty} \sim \vec{\pi}\right]$$

These functions allow agents to evaluate policies and guide learning toward long-term reward maximization.

Equilibrium and Solution Concepts

In game theory, equilibria represent stable strategy profiles where no agent has an incentive to deviate unilaterally. In MARL, equilibrium concepts are used to determine whether learning dynamics lead to such stable outcomes.

The most well-known concept is the **Nash equilibrium**, where each agent plays a best response to the strategies of others. However, MARL research also employs dynamic alternatives such as **regret minimization** and **reinforcement learning algorithms** like Q-learning.

Nash equilibrium is foundational for modeling strategic behavior (Hu and Wellman 1998), but in practice, MARL agents often approximate equilibrium behavior through learning algorithms that minimize regret over time or update action-values toward optimality.

Regret Minimization This process is illustrated in Figure 2

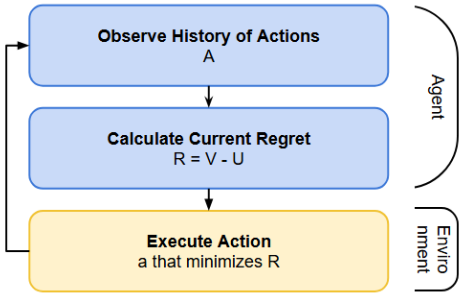

Figure 1: Regret minimization cycle: The agent observes action history, calculates regret for alternative strategies, and updates its policy to minimize cumulative regret.

At a high level, regret can be defined as the difference between the value an agent could have obtained by consistently playing its best fixed action in hindsight, and the value it actually obtained by following its chosen strategy.

This is commonly written as:

$$\bar{R}_i^{a_i}(T) = \bar{V}_i^{a_i}(T) - \bar{U}_i(T)$$

Formally:

$$\bar{U}_i(T) = \frac{1}{T}\sum_{t=1}^{T} u_i(a^t), \quad \bar{V}_i^{a_i}(T) = \frac{1}{T}\sum_{t=1}^{T} u_i(a_i, a_{-i}^t)$$

Where:

- i indexes a specific agent, with $i \in \{1, 2, \ldots, n\}$ for a system of n agents,
- T is the total number of time steps or learning rounds,

- t is the index of each round, ranging from 1 to T,
- a^t is the joint action profile at time t, i.e., $a^t = (a_1^t, a_2^t, \ldots, a_n^t)$,
- a_{-i}^t is the joint action of all agents other than agent i at time t,
- $u_i(\cdot)$ is the utility function for agent i.

Minimizing regret such that $\bar{R}_i^{a_i}(T) \to 0$ ensures agents converge to strategies that are, in hindsight, nearly optimal. This principle underlies algorithms like regret matching and no-regret dynamics (Jason R. Marden 2007; Cesa-Bianchi and Lugosi 2006; Loomes and Sugden 1982).

Q-Learning and Its Extensions Q-Learning's decision-making process is diagrammed in Figure 2.

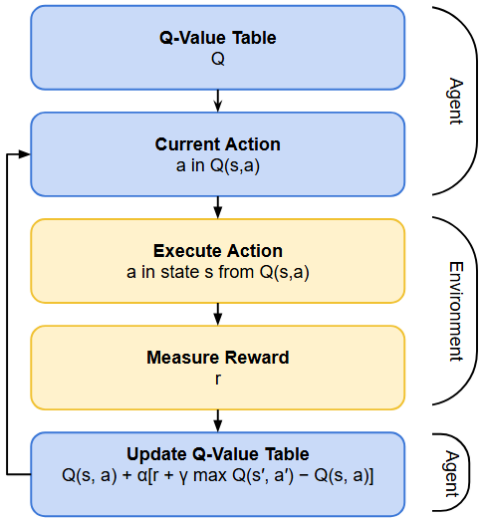

Figure 2: Q-learning process: The agent selects actions from its Q-table, interacts with the environment, observes rewards, and updates Q-values based on the Bellman equation.

Q-learning estimates the expected value of taking an action in a given state and following the current policy thereafter. The update rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

Where:

- s and s' are current and next states,
- a and a' are the current and future actions,
- r is the reward received for (s,a),
- α is the learning rate, which determines how quickly the algorithm incorporates new information (higher α means faster adaptation, but less stability),
- γ is the discount factor, which determines how much future rewards are valued relative to immediate ones (higher γ places more weight on long-term outcomes).

In multi-agent Q-learning, agents often optimize the joint action-value:

$$\vec{a}^* = \arg\max_{\vec{a}} \sum_{i=1}^{n} Q_i(s,\vec{a})$$

Where:

- n is the number of agents in the system,
- i indexes an individual agent, with $i \in \{1,2,\ldots,n\}$,
- $Q_i(s,\vec{a})$ is the Q-value estimated by agent i for taking joint action $\vec{a}$ in state s,
- $\arg\max_{\vec{a}}$ denotes the selection of the joint action that maximizes the sum of Q-values across all agents (typically used in fully cooperative settings).

This formulation is common in cooperative MARL. Variants like Nash Q-Learning, Friend-or-Foe Q-Learning, and Deep Q-Learning extend these principles to adversarial and mixed-motive environments (Hu and Wellman 1998; Jang et al. 2019; Foerster et al. 2016). This system is diagrammed at a high level in Figure 2

While Nash equilibrium provides a fixed-point solution in which each agent plays a best response to the strategies of others, regret minimization offers a dynamic alternative. In regret-based learning, agents iteratively adjust their strategies based on past performance to reduce cumulative regret. In reinforcement learning-based settings, convergence to Nash equilibrium is not guaranteed, but regret-minimizing and Q-learning-based methods often yield behavior that approximates equilibrium over time (Jason R. Marden 2007; Cesa-Bianchi and Lugosi 2006; Hu and Wellman 1998).

History

The conceptual foundations of reinforcement learning (RL) trace back to the behavioral psychology of Edward Thorndike, who in 1898 conducted experiments on animals learning to escape puzzle boxes via trial and error (Thorndike 1898). These experiments inspired early ideas about reward-based learning that later shaped machine learning frameworks.

In computing, Marvin Minsky proposed one of the first reinforcement-based learning systems in 1954 and implemented a simple learning machine in 1961 (MINSKY 1954). However, reinforcement learning did not gain widespread attention in AI until the 1980s and 1990s, when key mathematical frameworks were developed.

A major milestone came with the introduction of **Q-learning** by Watkins and Dayan in 1992, which provided a model-free method for agents to learn optimal policies by estimating action-values using temporal-difference updates (Watkins and Dayan 1992). This formulation became foundational for many subsequent learning algorithms and still underpins modern deep RL systems.

Multi-Agent Reinforcement Learning (MARL) has historically co-evolved with single-agent reinforcement learning. Even in the earliest experiments—such as Thorndike's puzzle box studies involving multiple cats—there was an implicit interest in how learning agents interact in shared environments. As reinforcement learning matured, MARL

emerged not as an entirely separate field, but as a natural extension that adapted single-agent techniques to multi-agent contexts. A key early contribution came from Hu and Wellman (1998), who formalized a MARL framework that incorporated game-theoretic equilibria, such as Nash Q-learning (Hu and Wellman 1998). Their work helped bridge classical game theory with learning dynamics in interactive, multi-agent environments.

As research progressed, various solution concepts were introduced to model agent behavior more realistically. Marden et al. (2007) introduced regret-based dynamics for weakly acyclic games, helping establish learning algorithms that could converge in more complex strategic environments (Jason R. Marden 2007). Around the same time, Tuyls and Parsons (2007) applied evolutionary game theory to MARL, studying population-level learning and adaptation (Tuyls and Parsons 2007).

By the 2010s, advances in function approximation and neural networks led to deep reinforcement learning (Deep RL), which made it possible to scale learning to high-dimensional environments. A landmark achievement was the development of **AlphaGo** by DeepMind, which used a combination of policy/value networks, Monte Carlo Tree Search (MCTS), and self-play reinforcement learning to master the game of Go (Silver et al. 2016). While AlphaGo involved a single agent, it simulated and learned from multi-agent interactions during self-play, making it an indirect success of MARL techniques.

Following AlphaGo, large-scale systems like OpenAI Five (for Dota 2) and DeepMind's AlphaStar (for StarCraft II) demonstrated MARL in explicitly multi-agent, partially observable, and complex domains. These systems required innovations in communication protocols, centralized training with decentralized execution, and scalable Q-learning variants (Vinyals et al. 2019; OpenAI et al. 2019; Foerster et al. 2016).

Recent survey papers (Yang and Wang 2020; Sven and Klaus 2022) highlight the co-evolution of reinforcement learning and game theory, and emphasize the importance of MARL in domains ranging from autonomous vehicles and smart grids to strategic trading systems. With advances in hardware and simulation environments, MARL has evolved from theoretical curiosity to a practical and essential framework for modeling complex agent interactions.

Results

One of the most famous and influential results in Multi-Agent Reinforcement Learning (MARL) is the success of **AlphaGo** in 2016, developed by DeepMind (Silver et al. 2016). AlphaGo was the first AI system to defeat a world champion Go player, and later several top-ranked players globally. This achievement marked a turning point not only in public recognition of reinforcement learning but also in the validation of deep MARL techniques for complex, strategic tasks.

From a technical standpoint, AlphaGo is significant because it models gameplay as a two-agent Markov game. Through self-play, it simulates interactions between symmetric agents using stochastic policies π, value functions $V^{\pi}(s)$, and Monte Carlo Tree Search (MCTS) to approximate optimal joint actions. Instead of relying on exhaustive enumeration like earlier systems (e.g., IBM's *Deep Blue* (Campbell, Hoane, and hsiung Hsu 2002)), AlphaGo learned both policy and value functions using deep neural networks trained via reinforcement learning and supervised learning. The complexity of Go—with roughly 10^{170} board states and a branching factor of 361 (Tromp and Farnebäck 2007; British Go Association 2017)—makes brute-force strategies impractical. AlphaGo's success demonstrates that MARL agents can learn optimal or near-optimal policies even in massive, partially observable environments.

This is especially relevant to real-world systems that share these properties—such as robotic control, autonomous vehicle coordination, and dynamic network allocation—where the number of states and potential interactions among agents is effectively unbounded (Yang and Wang 2020; Buşoniu, Babuška, and De Schutter 2010).

Another representative result from the literature is the application of MARL to **user-centric radio access technology (RAT) selection**, where agents dynamically choose between different wireless communication protocols to maximize system-level throughput while avoiding interference (Caso et al. 2021). In this context, each agent represents a mobile device or access point that must make decisions under partial observability and decentralized control. The problem is modeled as a Markov game with mixed-motive interaction: while all devices benefit from reduced interference, each also seeks to maximize its individual performance.

The study formalizes joint policy learning using Q-value approximations and introduces a distributed variant of multi-agent Q-learning with context-awareness and regret minimization. One key insight is that MARL systems can self-organize without explicit coordination when provided with local feedback and structured reward functions. This finding is significant because it addresses a practical, large-scale deployment problem in telecommunications—coordination without central control or full observability.

While AlphaGo remains a landmark demonstration of MARL's potential, more recent results like the radio access case illustrate how MARL frameworks can be generalized beyond games. According to recent surveys (Sven and Klaus 2022), many applications remain in the research phase, but the underlying algorithms—centralized training with decentralized execution, policy gradient methods, and deep Q-learning—continue to show promise across domains.

These results emphasize the importance of not only improving agent performance but also developing general benchmarks and evaluation protocols to compare different MARL systems in real-world settings.

Conclusion

As I explored the current state of Multi-Agent Reinforcement Learning (MARL), it became increasingly clear that while deep learning dominates modern AI development, effectively combining it with MARL introduces significant theoretical and practical challenges. One such challenge lies in the complexity of multi-agent environments themselves—characterized by non-stationarity, partial observ-

ability, and the need for coordination across agents with varying, and sometimes conflicting, objectives.

The success of AlphaGo (Silver et al. 2016) demonstrated that self-play among symmetric agents could yield superhuman performance in competitive, two-player zero-sum games. However, this competitive interaction type is only one part of the MARL landscape. Real-world systems are often better modeled as cooperative or mixed-motive settings, where agents either share a common objective or must balance collaboration with individual strategy. In these cases, the complexity of learning increases substantially due to issues such as credit assignment, decentralized coordination, and the need for emergent communication protocols (Yang and Wang 2020; Sven and Klaus 2022).

While MARL has demonstrated remarkable capabilities in strategic, interactive environments, not all breakthrough AI systems depend on multi-agent frameworks. For instance, **AlphaFold**—DeepMind's protein structure prediction model—achieved state-of-the-art results by predicting protein folding structures with near-experimental accuracy (Jumper et al. 2021; Bryant, Pozzati, and Elofsson 2022). Notably, AlphaFold relies primarily on deep supervised learning and attention mechanisms, without incorporating reinforcement learning or multi-agent coordination. This distinction highlights an important boundary: in domains with well-defined objectives, abundant labeled data, and clear evaluation metrics, single-agent architectures may be more efficient and scalable than MARL systems. As such, AlphaFold serves as a counterpoint to MARL successes like AlphaGo, and invites deeper inquiry into when and why multi-agent learning is the appropriate paradigm.

Looking forward, one promising research direction is the development of **specialized, cooperative agents** that contribute to a shared task through differentiated roles—akin to players on a sports team or workers in a distributed organization. This idea was explored by OpenAI in its Dota 2 system, where agents executed distinct roles to achieve team-level goals (OpenAI et al. 2019). These systems align with the cooperative or mixed-motive interaction types defined earlier and push the boundaries of what MARL can achieve in partially observable, multi-agent domains.

This specialization-based approach has direct implications for my own research. I am currently building an intelligent assistant composed of modular agents—for instance, one handling calendar management, another responsible for facial recognition, and a third interpreting speech. Each agent has a different observation space, action set, and reward function, yet they all contribute toward a common goal. This raises a core question in cooperative MARL: *How can we evaluate and align these heterogeneous agents to ensure effective joint behavior?*

From a modeling perspective, addressing this challenge may require extending the Markov game framework to incorporate heterogeneous reward structures and task hierarchies. New formulations could involve hierarchical MARL models, where higher-level policies allocate subgoals to specialized agents. These architectures would need to solve the coordination problem not just across agents, but across abstract task layers—exacerbating issues of scalability, communication, and credit assignment.

Solving this evaluation and coordination problem could unlock a new class of MARL applications in domains like human-AI collaboration, distributed robotics, and AI-assisted memory systems. As the field matures, I believe that addressing these foundational coordination challenges—especially in cooperative and mixed-motive settings—will be crucial to realizing the full potential of MARL.

References

Alonso, M. N. I.; and Njupoun, A. M. 2024. Game Theory and Multi-Agent Reinforcement Learning: A Mathematical Overview. Technical report, SSRN. Accessed: March 30, 2025.

British Go Association. 2017. A Comparison of Chess and Go. Published by the British Go Association.

Bryant, P.; Pozzati, G.; and Elofsson, A. 2022. Improved prediction of protein-protein interactions using AlphaFold2. *Nature Communications*, 13(1): 1265.

Busoniu, L.; Babuska, R.; and De Schutter, B. 2006. Multi-Agent Reinforcement Learning: A Survey. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, 1–6.

Buşoniu, L.; Babuška, R.; and De Schutter, B. 2010. *Multi-agent Reinforcement Learning: An Overview*, 183–221. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-14435-6.

Campbell, M.; Hoane, A.; and hsiung Hsu, F. 2002. Deep Blue. *Artificial Intelligence*, 134(1): 57–83.

Caso, G.; Alay, ; Ferrante, G. C.; Nardis, L. D.; Benedetto, M.-G. D.; and Brunstrom, A. 2021. User-Centric Radio Access Technology Selection: A Survey of Game Theory Models and Multi-Agent Learning Algorithms. *IEEE Access*, 9: 84417–84464.

Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, Learning, and Games*. Cambridge University Press. ISBN 9780521841085. Chapter 2 contains formal regret definitions and derivations.

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2137–2145. Accessed: March 30, 2025.

Hu, J.; and Wellman, M. P. 1998. Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, 242–250. Accessed: March 30, 2025.

Jang, B.; Kim, M.; Harerimana, G.; and Kim, J. W. 2019. Q-Learning Algorithms: A Comprehensive Classification and Applications. *IEEE Access*, 7: 133653–133667.

Jason R. Marden, J. S. S., Gurdal Arslan. 2007. Regret Based Dynamics:Convergence in Weakly Acyclic Games. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 194–201. ACM. Accessed: March 30, 2025.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; and Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873): 583–589.

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; and Graepel, T. 2017. A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 4190–4203. Accessed: March 30, 2025.

Loomes, G.; and Sugden, R. 1982. Regret Theory: An Alternative Theory of Rational Choice Under Uncertainty. *The Economic Journal*, 92(368): 805–824.

MINSKY, M. L. 1954. *THEORY OF NEURAL-ANALOG REINFORCEMENT SYSTEMS AND ITS APPLICATION TO THE BRAIN-MODEL PROBLEM*. Ph.D. thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-02-17.

OpenAI; :; Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; Józefowicz, R.; Gray, S.; Olsson, C.; Pachocki, J.; Petrov, M.; d. O. Pinto, H. P.; Raiman, J.; Salimans, T.; Schlatter, J.; Schneider, J.; Sidor, S.; Sutskever, I.; Tang, J.; Wolski, F.; and Zhang, S. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. arXiv:1912.06680.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Sven, G.; and Klaus, D. 2022. Multi-agent deep reinforcement learning: a survey. *The Artificial Intelligence Review*, 55(2): 895–943. Copyright - © The Author(s) 2021. This work is published under http://creativecommons.org/licenses/by/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2024-03-22.

Tan, M. 1993. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Machine Learning Proceedings 1993*, 330–337. San Francisco (CA): Morgan Kaufmann. ISBN 978-1-55860-307-3.

Thorndike, E. L. 1898. Review of Animal Intelligence: An Experimental Study of the Associative Processes in Animals. *Psychological Review*, 5(5): 551–553. Review of the book *Animal Intelligence: An Experimental Study of the Associative Processes in Animals* by E. L. Thorndike.

Tromp, J.; and Farnebäck, G. 2007. Combinatorics of Go. In van den Herik, H. J.; Ciancarini, P.; and Donkers, H. H. L. M. J., eds., *Computers and Games*, 84–99. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-75538-8.

Tuyls, K.; and Parsons, S. 2007. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7): 406–416. Foundations of Multi-Agent Learning.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Watkins, C. J. C. H.; and Dayan, P. 1992. Q-learning. *Machine Learning*, 8(3): 279–292.

Yang, Y.; and Wang, J. 2020. An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective. *arXiv preprint arXiv:2011.00583*. Accessed: March 30, 2025.

Zhang, K.; Yang, Z.; and Başar, T. 2021. 321–384. Cham: Springer International Publishing. ISBN 978-3-030-60990-0.