Barcelona School of Economics

Resource Competition in Multi-Agent Reinforcement Learning: A Literature Review

Reinforcement Learning - DSDM

Anastasia Chernavskaia

Moritz Peist

Nicolas Rauth

Abstract

Multi-agent reinforcement learning in competitive resource environments represents a critical domain where autonomous agents must learn optimal strategies while competing for limited resources. This literature review examines recent developments in competitive multi-agent systems, with particular emphasis on resource allocation scenarios. We survey foundational work including OpenAI's MADDPG framework, sequential social dilemmas, emergent complexity through competition, and practical applications across industries. The review identifies key algorithmic approaches, addresses fundamental challenges, including non-stationarity and credit assignment in competitive settings, and highlights current research directions toward more robust and scalable competitive multi-agent systems.

June 30, 2025

Contents

1 Introduction

Multi-agent reinforcement learning (MARL) in competitive resource environments presents unique challenges where agents must simultaneously learn optimal policies while adapting to the evolving strategies of competing agents. Unlike purely cooperative or zero-sum scenarios, resource competition introduces complex dynamics where agents compete for scarce resources while potentially benefiting from limited cooperation (Lowe, Wu, et al., 2017).

The significance of this domain extends beyond theoretical interest to practical applications in cloud computing, wireless networks, autonomous vehicles, and financial markets, where multiple autonomous entities must efficiently allocate limited resources while pursuing individual objectives (Allahham et al., 2022; Wang et al., 2021). These environments exhibit fundamental properties that distinguish them from single-agent learning: non-stationarity from each agent's perspective, scalability challenges with increasing agent populations, and emergent behaviors that arise from competitive interactions (Hernandez-Leal et al., 2019; Ning & Xie, 2024).

2 Foundational Frameworks and Algorithmic Approaches

The development of practical algorithms for competitive resource environments has required fundamental innovations in multi-agent learning architectures. The core challenge lies in addressing non-stationarity while enabling agents to learn effective competitive strategies. This section examines the foundational algorithmic contributions that have shaped the field, beginning with centralized training approaches that maintain decentralized execution, followed by value decomposition methods and policy gradient techniques adapted explicitly for competitive scenarios.

2.1 MADDPG and Centralized Training with Decentralized Execution

The seminal work by Lowe, Wu, et al. (2017) introduced Multi-Agent Deep Deterministic Policy Gradient (MADDPG), establishing a foundational framework for learning in mixed cooperative-competitive environments. MADDPG addresses the fundamental challenge of non-stationarity by employing centralized training with decentralized execution (CTDE), where agents use centralized critics during training that have access to all agents' observations and actions, while maintaining decentralized policies for execution.

The algorithm extends single-agent DDPG to multi-agent settings through a key innovation: each agent i maintains its own actor network $\mu_i(o_i|\theta_i)$ but uses a centralized critic $Q_i^\mu(x, a_1, \ldots, a_N)$ where x represents the global state and $a_1, \ldots, a_N$ are all agents' actions. The policy gradient for agent i becomes:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}[\nabla_{\theta_i} \mu_i(o_i) \nabla_{a_i} Q_i^\mu(x, a_1, \ldots, a_N)|_{a_i=\mu_i(o_i)}] \tag{1}$$

This approach demonstrates superior performance compared to independent learning ap-

proaches in environments featuring resource competition, such as the "simple tag" scenario, where agents compete to reach specific locations while avoiding adversaries (cf. our implementation in A). The OpenAI experiments revealed critical insights about how resource availability affects agent behavior and learning dynamics, establishing that multi-agent environments possess inherent curriculum properties where increasing agent capabilities create correspondingly more challenging environments (Lowe, Mordatch, et al., 2017).

2.2 Value Decomposition and Policy Gradient Methods

For competitive resource allocation, value decomposition methods address the credit assignment problem while maintaining competitive dynamics. QMIX (Rashid et al., 2020) and its variants enable agents to learn joint action-value functions through monotonic mixing networks, ensuring that individual Q-value maximization leads to optimal joint actions in cooperative sub-tasks while maintaining competitive behavior where appropriate. Recent advances include QPLEX (Wang et al., 2021), which introduces a duplex dueling architecture for Q-function decomposition, and Weighted QMIX, which breaks monotonicity limitations to handle more complex competitive interactions.

Furthermore, multi-Agent Proximal Policy Optimization (MAPPO) (Yu et al., 2022) has demonstrated surprising effectiveness in competitive environments despite its conceptual simplicity, with success stemming from the proper implementation of centralized value functions rather than algorithmic complexity. Counterfactual Multi-Agent Policy Gradients (COMA) (Foerster et al., 2018) addresses credit assignment in competitive settings through counterfactual baselines that measure individual agent contributions while marginalizing out other agents' actions.

3 Sequential Social Dilemmas and Emergent Complexity

The evolution from simple matrix games to complex temporal environments has revealed fundamental insights about how resource competition shapes agent behavior over extended time horizons. This section explores how sequential social dilemmas capture the essence of real-world resource competition, where agents must balance immediate gains against long-term strategic positioning. We examine both the theoretical foundations of these environments and the remarkable complexity that emerges through competitive self-play, demonstrating how simple competitive pressure can give rise to sophisticated behavioral patterns.

3.1 Temporal Resource Competition

Leibo et al. (2017) extended the traditional matrix game framework to sequential social dilemmas (SSDs), recognizing that real-world resource competition involves temporally extended decisions rather than single-shot actions. Their work introduced paradigmatic environments that capture essential aspects of resource competition, including the Fruit

Gathering Game where agents navigate spatial environments to collect finite fruit resources, and the Wolfpack Hunting Game that models predator coordination balancing individual reward maximization with collective hunting benefits.

The SSD framework reveals how environmental factors, particularly resource abundance, fundamentally shape learned behaviors. When resources are plentiful, agents pursue individual strategies with minimal interference. As scarcity increases, emergent phenomena occur including territorial establishment and defense, sophisticated temporal strategies for resource acquisition, and adaptive counter-strategies in response to opponents' evolving policies. These findings demonstrate that cooperativeness is a property of policies rather than elementary actions, requiring agents to learn complex behavioral patterns balancing immediate rewards with long-term strategic positioning.

3.2 Autocurricula and Behavioral Complexity

Bansal et al. (2018) demonstrated that competitive multi-agent environments trained with self-play can produce behaviors far more complex than the environment itself. Their work in 3D physics-based competitive environments shows that agents develop sophisticated skills including running, blocking, tackling, and strategic deception through purely competitive pressure. The concept of autocurricula emerges as a critical feature, where agents' performance improvements effectively modify the learning environment for all participants, creating stacked layers of learning where each behavioral innovation provides the foundation for the next level of complexity.

This research identifies several mechanisms through which competitive pressure generates behavioral complexity, including arms race dynamics that create continuous pressure for counter-strategy development, natural curriculum adjustment that matches challenge levels to agent capabilities, skill transfer from competitive contexts to related tasks, and emergent tool use where agents discover and exploit environmental features for competitive advantage. The feedback loop ensures that, regardless of skill level, the environment contains opponents that are appropriately challenging.

4 Industry Applications and Practical Implementations

The transition from laboratory environments to real-world deployment has demonstrated the practical viability of competitive multi-agent learning across diverse industries. This section examines how theoretical advances in competitive multi-agent reinforcement learning (MARL) have been successfully applied to solve concrete resource allocation problems in cloud computing, wireless networks, and financial markets. These applications reveal both the potential and the practical challenges of deploying competitive agents in environments where resource constraints drive natural competition among autonomous systems.

4.1 Cloud Computing and Network Resource Management

Belgacem et al. (2022) developed the Intelligent Multi-Agent Reinforcement Learning Model (IMARM) for cloud resource allocation, where virtual machines compete for computational resources while optimizing energy consumption and fault tolerance. The multi-agent approach enables dynamic resource allocation responding to changing consumer demands, with Q-learning policies guiding virtual machines to optimal states based on current environmental conditions. Competitive elements arise from finite computational resources and the need to balance multiple objectives, including energy efficiency, load balancing, and quality of service.

Recent work in 6G wireless networks (Zhang et al., 2025) employs MARL for dynamic spectrum allocation and network selection, where edge nodes compete for radio access technologies and bandwidth resources. The multi-agent framework addresses natural network decentralization while enabling intelligent coordination. Key competitive elements include spectrum competition among multiple nodes for limited frequency bands, quality of service optimization that balances individual performance with network-wide efficiency, and energy management that optimizes battery lifetime while maintaining service quality.

4.2 Financial Markets and Resource Competition

Multi-agent systems in financial markets represent natural competitive resource environments where trading agents compete for profitable opportunities in markets with limited liquidity (Shavandi & Khedmati, 2022). Recent developments employ MARL for algorithmic trading, where agents learn to adapt to market conditions and competitor strategies. This includes portfolio optimization, which involves dynamic capital allocation across multiple assets, and risk management, balancing individual returns with systemic risk considerations. The competitive nature arises from zero-sum trading aspects where one agent's profit often corresponds to another's loss, combined with finite liquidity available at any price level.

In competitive resource environments, communication presents challenges where agents may benefit from coordination to avoid wasteful conflicts while maintaining competitive advantages. Recent work explores learned communication protocols that balance information sharing with competitive considerations (Foerster et al., 2016), strategic information sharing that maximizes individual utility while enabling beneficial coordination (Eccles et al., 2019), and deceptive communication, where agents learn to provide misleading information for competitive advantages (Cao et al., 2018).

5 Current Challenges and Future Research Directions

Despite significant progress in competitive multi-agent learning, several fundamental challenges remain that limit the scalability and practical deployment of these systems. This section addresses the primary obstacles facing the field, including computational scalability as agent populations grow, robustness against adversarial exploitation, and the need for stronger theoretical foundations. We examine current research approaches to these

challenges and identify promising directions for future investigation that could enable more robust and scalable competitive multi-agent systems.

5.1 Scalability and Robustness

Current MARL approaches face significant scalability challenges as the number of competing agents increases, with joint action spaces growing exponentially and centralized training becoming computationally prohibitive. Research directions include mean-field approaches that approximate large-scale interactions through average effects, graph neural networks that leverage structural relationships for efficient computation, and hierarchical methods that decompose large-scale problems into manageable sub-problems (Liu et al., 2024; Ma et al., 2024).

Competitive environments pose unique robustness challenges as adversarial interactions can exploit learned vulnerabilities. Current research focuses on population-based training, maintaining diverse agent populations to improve robustness, adversarial training that explicitly targets worst-case opponents, and domain randomization, which enhances generalization across varied competitive scenarios (Zhou et al., 2023).

5.2 Theoretical Foundations and Evaluation

The theoretical analysis of competitive resource allocation builds on game-theoretic foundations, particularly the existence and computation of Nash equilibria in multi-agent settings. However, traditional equilibrium concepts often fail to capture dynamic learning in resource-constrained environments. Recent theoretical work explores correlated equilibria as more robust solution concepts for multi-agent learning, as well as no-regret learning algorithms with theoretical guarantees in competitive settings, and theoretical frameworks for opponent modeling that facilitate learning about and adapting to competitor strategies (Fuente et al., 2024).

Independent learning in competitive environments typically lacks convergence guarantees due to non-stationarity. Research addresses this issue through multi-timescale learning, which utilizes different learning rates to stabilize competitive dynamics, best response dynamics, and theoretical analysis of agent response patterns, as well as regret minimization algorithms that minimize worst-case performance against adaptive opponents.

The competitive MARL community lacks standardized benchmarks that capture the full complexity of resource competition. Recent efforts include PettingZoo environments, which provide standardized multi-agent environments with competitive scenarios, Melting Pot, DeepMind's evaluation suite for social interaction, including resource competition, and JaxMARL, offering high-performance implementations that enable large-scale experimentation (Liu et al., 2024).

6 Conclusion

Multi-agent reinforcement learning in competitive resource environments has evolved from foundational frameworks, such as MADDPG, to sophisticated applications across various

industries, including cloud computing, wireless networks, and financial markets. The field has demonstrated that resource scarcity fundamentally shapes emergent behaviors, leading to complex strategic interactions extending far beyond simple competition.

Key contributions include establishing sequential social dilemmas as a paradigm for understanding temporal resource competition, demonstrating emergent complexity through competitive pressure, and developing scalable algorithms for real-world resource allocation problems. Current challenges focus on scalability, robustness, and theoretical understanding of learning dynamics in competitive settings.

Future research directions emphasize the development of more sophisticated evaluation frameworks, improving theoretical guarantees for learning in competitive environments, and exploring hybrid cooperative-competitive scenarios that better reflect real-world resource allocation challenges. The integration of large language models and foundation model approaches presents new opportunities for developing more capable and generalizable competitive agents. For instance, Fish et al. (2025) demonstrated collusion in multi-agent environments under specific prompt settings, where large language models competing in a fictional market created new coordination challenges rather than genuine competition.

The field's evolution toward practical deployment demonstrates the maturity of competitive multi-agent reinforcement learning (MARL) as a technology for addressing real-world resource allocation challenges, while continuing to provide insights into the fundamental nature of multi-agent learning and strategic interaction.

References

Allahham, M. S., Abdellatif, A. A., Mhaisen, N., Mohamed, A., Erbad, A., & Guizani, M. (2022). Multi-agent reinforcement learning for network selection and resource allocation in heterogeneous multi-RAT networks. *IEEE Transactions on Cognitive Communications and Networking*, *8*(2), 1287–1300. https://doi.org/10.1109/TCCN.2022.3155727

Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., & Mordatch, I. (2018, March 14). Emergent complexity via multi-agent competition. https://doi.org/10.48550/arXiv.1710.03748

Belgacem, A., Mahmoudi, S., & Kihl, M. (2022). Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 2391–2404. https://doi.org/10.1016/j.jksuci.2022.03.016

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., & Clark, S. (2018, April 11). Emergent communication through negotiation. https://doi.org/10.48550/arXiv.1804.03980

Eccles, T., Bachrach, Y., Lever, G., Lazaridou, A., & Graepel, T. (2019). Biases for emergent communication in multi-agent reinforcement learning. In *Proceedings of the 33rd international conference on neural information processing systems* (pp. 13121–13131). Curran Associates Inc.

Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2025). Algorithmic collusion by large language models. *AEA Paper Session: AI-Driven Market Dynamics*. https://doi.org/10.48550/arXiv.2404.00806

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2145–2153.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi-agent policy gradients. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2974–2982.

Fuente, N. D. L., Alonso, M. N. i., & Casadellà, G. (2024, December 29). Game theory and multi-agent reinforcement learning : From nash equilibria to evolutionary dynamics. https://doi.org/10.48550/arXiv.2412.20523

Hernandez-Leal, P., Kartal, B., & Taylor, M. E. (2019). A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, *33*(6), 750–797. https://doi.org/10.1007/s10458-019-09421-1

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., & Graepel, T. (2017). Multi-agent reinforcement learning in sequential social dilemmas. *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 464–473.

Liu, D., Ren, F., Yan, J., Su, G., Gu, W., & Kato, S. (2024). Scaling up multi-agent reinforcement learning: An extensive survey on scalability issues. *IEEE Access*, *12*, 94610–94631. https://doi.org/10.1109/ACCESS.2024.3410318

Lowe, R., Mordatch, I., Abbeel, P., Wu, Y., Tamar, A., & Harb, J. (2017, June 7). *Learning to cooperate, compete, and communicate*. Retrieved June 23, 2025, from https://openai.com/index/learning-to-cooperate-compete-and-communicate/

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6382–6393.

Ma, C., Li, A., Du, Y., Dong, H., & Yang, Y. (2024). Efficient and scalable reinforcement learning for large-scale network control [Publisher: Nature Publishing Group]. *Nature Machine Intelligence*, *6*(9), 1006–1020. https://doi.org/10.1038/s42256-024-00879-7

Ning, Z., & Xie, L. (2024). A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, *3*(2), 73–91. https://doi.org/10.1016/j.jai.2024.02.003

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., & Whiteson, S. (2020). Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.*, *21*(1), 178:7234–178:7284.

Shavandi, A., & Khedmati, M. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications*, *208*, 118124. https://doi.org/10.1016/j.eswa.2022.118124

Wang, J., Ren, Z., Liu, T., Yu, Y., & Zhang, C. (2021, October 4). QPLEX: Duplex dueling multi-agent q-learning. https://doi.org/10.48550/arXiv.2008.01062

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of PPO in cooperative multi-agent games. *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 24611–24624.

Zhang, J., Liu, Z., Zhu, Y., Shi, E., Xu, B., Yuen, C., Niyato, D., Debbah, M., Jin, S., Ai, B., Xuemin, & Shen. (2025, February 9). Multi-agent reinforcement learning in wireless distributed networks for 6g. https://doi.org/10.48550/arXiv.2502.05812

Zhou, M., Wan, Z., Wang, H., Wen, M., Wu, R., Wen, Y., Yang, Y., Yu, Y., Wang, J., & Zhang, W. (2023). MALib: A parallel framework for population-based multi-agent reinforcement learning. *J. Mach. Learn. Res.*, *24*(1), 150:7205–150:7216.

A Appendix

To provide empirical context for our literature review, we implemented a multi-agent coordination experiment using Proximal Policy Optimization (PPO) in two environments from the Multi-Agent Particle Environment (MPE) suite: Simple Spread (cooperation) and Simple Adversary (competition). These environments require multiple agents to coordinate their movements to cover distinct landmarks without collisions, while escaping from the adversary agent. These games represent a fundamental resource allocation challenge where spatial positions serve as contested resources.

A.1 Simple Spread Environment

The Simple Spread environment instantiates key challenges in competitive multi-agent learning. Three agents must simultaneously learn to occupy different landmarks while avoiding collisions with each other. Each agent observes its own position, velocity, and relative positions to landmarks and other agents. The reward structure encourages agents to cover landmarks while penalizing collisions, creating an implicit resource competition where optimal positioning requires coordination rather than pure competition.

A.2 Simple Adversary Environment

The Simple Adversary environment blends cooperation and competition by pitting two "good" agents (blue) against one adversary (red), with landmarks equal to the number of good agents. Each episode, only the good agents know which landmark is the target. They are rewarded for proximity to the target, while the adversary—unaware of the target's identity—must infer it from the good agents' behavior and is rewarded for approaching it. This setup forces good agents to coordinate and deceive, while the adversary predicts and intercepts. With observations including relative positions and discrete movement actions, the environment serves as a concise testbed for partial observability, deception, and adaptive opponent modeling in competitive multi-agent reinforcement learning.

These tasks exemplify the core challenges addressed in our literature review: agents must solve credit assignment problems (determining which agent should target which landmark), handle non-stationarity (as other agents' policies evolve during training), and develop coordination strategies that emerge from individual learning processes.

A.3 PPO Implementation

Our implementation follows the centralized training with decentralized execution paradigm, where agents share a critic network during training but execute policies independently. We track key metrics including episode rewards, episode lengths, value function losses, and policy gradient losses to analyze learning dynamics and convergence properties.

The experiment demonstrates empirically how multi-agent environments create natural curricula—as agents improve, the coordination challenge increases correspondingly. This observation directly motivates the theoretical frameworks examined in our literature review, particularly regarding emergent complexity in competitive resource environments

and the challenges of achieving stable learning in non-stationary multi-agent settings.

A.4 Neural Network Architecture

A.4.1 Policy Network Architecture

```
policy_kwargs = dict(
    net_arch=[256, 256, 128],
    activation_fn=torch.nn.Tanh)
```

Architecture Description:

- **Input Layer**: Flattened observation vectors from the environment (agent positions, velocities, relative distances)
- **Hidden Layer 1**: 256 neurons with Tanh activation
- **Hidden Layer 2**: 256 neurons with Tanh activation
- **Hidden Layer 3**: 128 neurons with Tanh activation
- **Output Layers**:
  - **Actor head**: Continuous action outputs (movement forces in x, y directions)
  - **Critic head**: Single value estimate for state evaluation

A.4.2 Activation Function Choice

- **Tanh activation**: Chosen for continuous control tasks as it provides bounded outputs $(-1, 1)$, naturally suited for movement commands in the adversarial environment
- **Smooth gradients**: Facilitates stable learning in competitive multi-agent settings

A.5 Training Methodology

A.5.1 Algorithm: Proximal Policy Optimization (PPO)

```
model = PPO(
    ActorCriticPolicy,
    env=env,
    learning_rate=3e-4,
    n_steps=256,
    batch_size=512,
    ent_coef=0.01,
    gamma=0.98,
    gae_lambda=0.95,
    verbose=1,
    tensorboard_log="./ppo_marl_tb/",
    policy_kwargs=policy_kwargs)
```

A.5.2 Hyperparameter Configuration

Learning Parameters:

- **Learning Rate**: 3e-4 (conservative rate for stable competitive learning)
- **Rollout Buffer**: 256 steps per update cycle
- **Batch Size**: 512 (large batches for reduced variance in competitive settings)

PPO-Specific Parameters:

- **Entropy Coefficient**: 0.01 (encourages exploration while maintaining policy stability)
- **Discount Factor (γ)**: 0.98 (values future rewards highly for strategic planning)
- **GAE Lambda**: 0.95 (Generalized Advantage Estimation for bias-variance trade-off)

A.5.3 Training Scale

- **Total Timesteps**: 500,000 (extended training for complex strategy development)
- **Training Updates**: $\sim$1,953 PPO updates (500,000 $\div$ 256 rollout buffer)
- **Episode Length**: Maximum 100 steps per episode

A.6 Multi-Agent Training Approach

A.6.1 Parameter Sharing Strategy

- **Centralized Training**: All agents (both good agents and adversaries) share the same policy network parameters
- **Decentralized Execution**: During evaluation, each agent acts independently based on local observations
- **Homogeneous Agents**: Identical network architecture for all agent types, with role differentiation emerging through reward structure

A.6.2 Environment Preprocessing Pipeline

```
env = ss.black_death_v3(env)                # Handle agent termination
env = ss.pad_observations_v0(env)           # Standardize obs dimensions
env = ss.flatten_v0(env)                    # Convert to vector format
env = aec_to_parallel(env)                  # Enable simultaneous actions
env = ss.pettingzoo_env_to_vec_env_v1(env) # Vectorization compatibility
```

A.6.3 Observation Processing

- **Input**: Multi-dimensional observations including agent positions, velocities, and relative distances

- **Flattening**: Converts structured observations to flat vectors suitable for MLP processing
- **Padding**: Ensures consistent input dimensions across different agent types
- **Normalization**: Handled implicitly through environment wrappers

A.6.4 Competitive Learning Dynamics

- **Zero-Sum Structure**: Good agents maximize rewards for reaching landmarks while avoiding capture; adversaries maximize rewards for catching good agents
- **Self-Play Training**: Agents improve by competing against increasingly sophisticated opponents
- **Emergent Complexity**: Strategic behaviors emerge through competitive pressure rather than explicit programming
- **Natural Curriculum**: Difficulty scales automatically as agents improve

A.6.5 Training Stability Considerations

- **Conservative Hyperparameters**: Lower learning rates and entropy coefficients prevent instability in non-stationary multi-agent environments
- **Large Batch Sizes**: Reduce variance in gradient estimates when multiple agents are learning simultaneously
- **Extended Training**: 500,000 timesteps allow for sophisticated strategy development and convergence

This architecture and training methodology enables the emergence of complex competitive behaviors such as coordinated pursuit strategies, evasion patterns, and spatial awareness through pure reinforcement learning optimization.

A.7 Connection to Literature Review

This practical implementation grounds our literature review in concrete multi-agent learning challenges by leveraging both the Simple Spread and Simple Adversary environments. The Simple Spread task encapsulates fundamental themes from the literature, such as resource competition for spatial positions, coordination under partial observability, and the emergence of complex group behaviors from simple, decentralized policies. In contrast, the Simple Adversary environment introduces an explicit adversarial element, requiring agents to balance cooperation with deception and opponent modeling as they navigate a setting where one agent seeks to thwart the goals of the others. The empirical training dynamics observed in our experiments—including convergence patterns, the development of coordinated strategies, and adaptive responses to adversarial pressure—provide a tangible context for understanding the theoretical contributions and algorithmic advances discussed in the literature review. By examining both cooperative and mixed-motive scenarios, our implementation illustrates how diverse multi-agent environments instantiate and challenge the core ideas explored in contemporary research.

The experiment's training results (Figure 1, Figure 2 and Figure 3) illustrate the learning curves and coordination development that motivate the need for sophisticated multi-agent algorithms, directly connecting our practical experience to the surveyed theoretical advances in competitive multi-agent reinforcement learning.

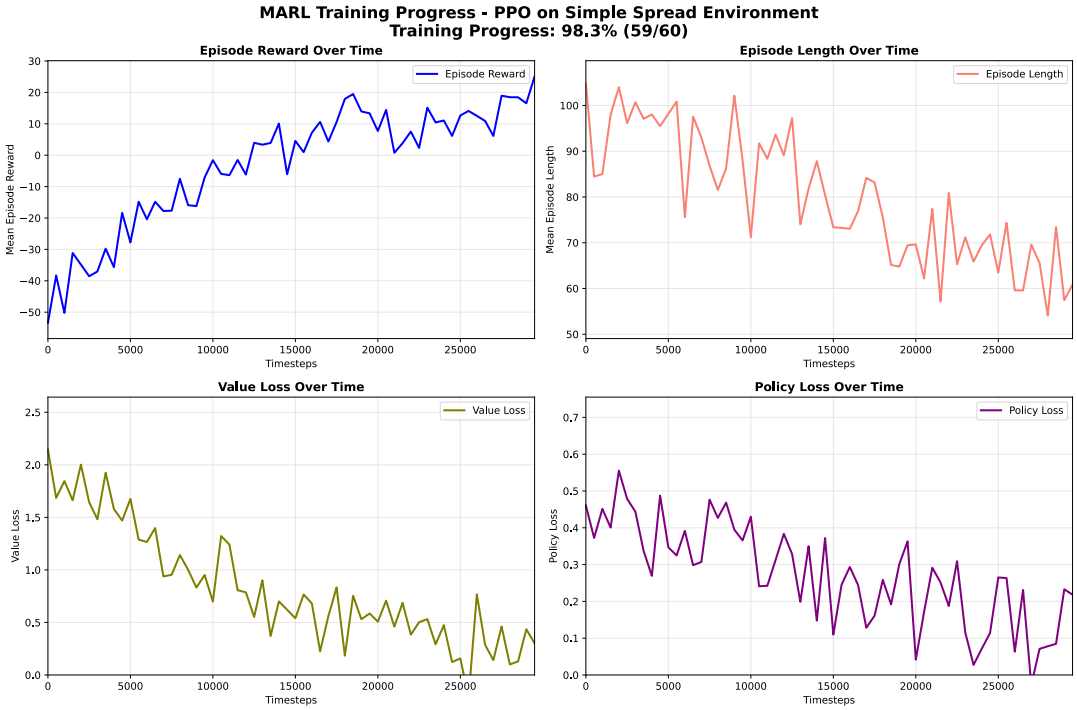

Figure 1: PPO training metrics on Simple Spread environment. Clockwise from top-left: episode rewards, episode lengths, policy losses, and value function losses over 30k timesteps.

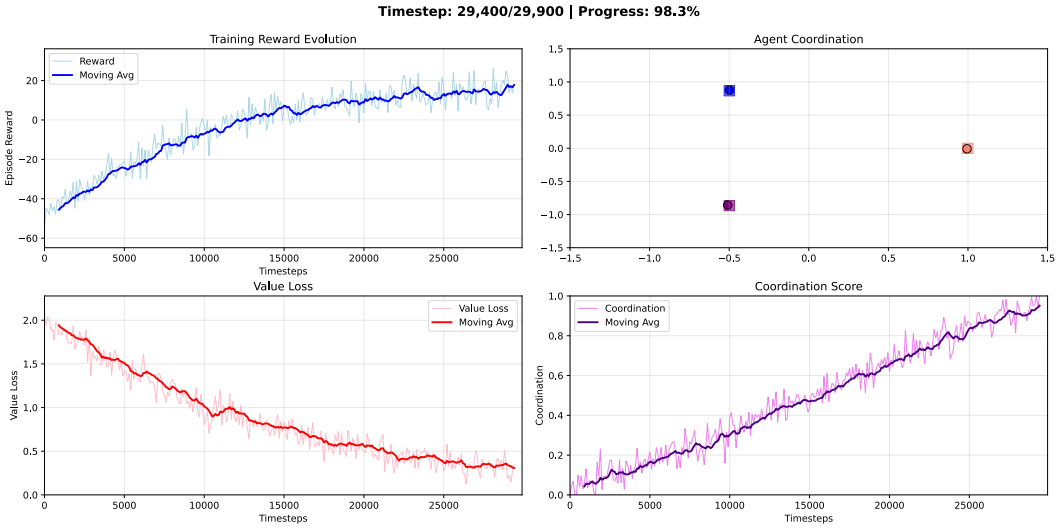

Figure 2: Multi-agent coordination dashboard. Clockwise from top-left: showing reward progression, agent behaviors, value loss, and coordination score during PPO training.

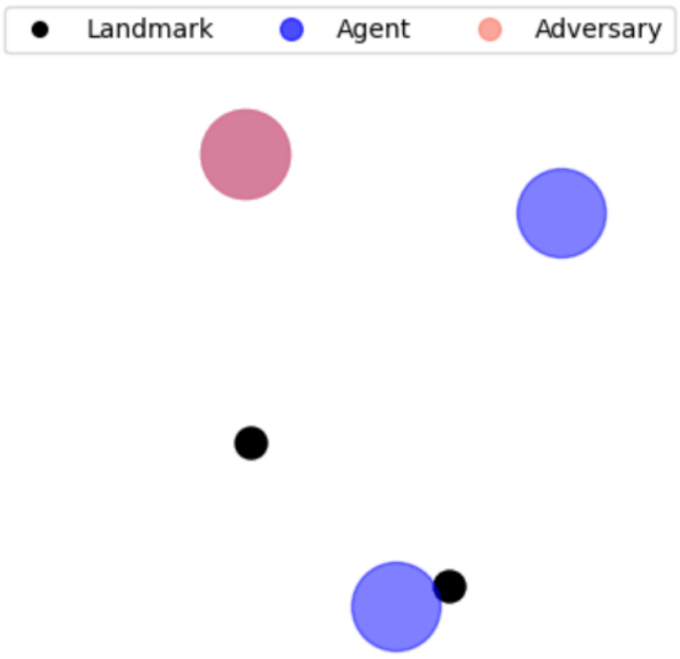

Figure 3: Screenshot from Simple Adversary training process. The blue "good" agents have to reach a landmark and escape the red adversary.