REPORT

Capture The Flag (CTF) using Multi Agent RL Algorithms

Gavit Deepesh Ravikant

20114

Nov 25

Teammates: Shirish Jha (2410705), Mohit Wadhwa (22205)

Overview

- Problem Statement
- Algorithms
- Methodology
- Results
- Conclusion

Introduction

- This report explores the application of Multi-Agent Reinforcement Learning (MARL) algorithms in a simulated Capture The Flag (CTF) environment. Specifically, it compares the performance of Independent Q-Learning (IQL) and Multi-Agent Proximal Policy Optimization (MAPPO) in handling coordination, strategy formation, and adaptability.
- The study reveals that MAPPO outperforms IQL in dynamic and cooperative settings due to its ability to foster inter-agent coordination, while IQL performs adequately in simpler, less dynamic environments.

Motivation

- The primary motivation of this study is to investigate how different MARL algorithms handle the challenges posed by a competitive and dynamic environment like CTF.
- By comparing Independent Q-Learning (IQL) and Multi-Agent Proximal Policy Optimization (MAPPO), we aim to understand the strengths and limitations of each algorithm in terms of coordination, strategy formation, and adaptability.

Problem Statement

- Capture the Flag environment has two teams with two agents in each team.
- Every team has the objective of capturing the opponent's flag, but at the same time defend its own.
- Defending the flag activates when an agent enters a visual depth of 3 near the opponent's flag.
- Obstacles, and flags positions were static, and two agents could occupy same cell.

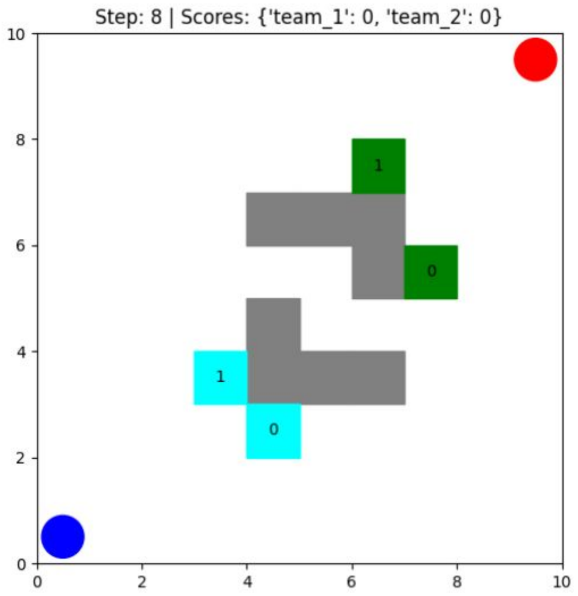


Figure: CTF Environment

Algorithms Used:

- **IQL:** Independent Q Learning
- **MAPPO:** Multi Agent Proximal Policy Optimization

IQL: Independent Q Learning

IQL treats each agent as an independent learner. Each agent individually estimates Q-values and updates its policy based on personal experiences without considering the actions or policies of other agents.

Advantages

- Simplicity: Straightforward to implement and understand.
- Scalability: Works well in environments where agents operate independently.

Limitations

- Non-Stationarity: The environment appears non-stationary to each agent due to the actions of others, leading to convergence issues.
- Lack of Coordination: Agents cannot develop cooperative strategies, limiting effectiveness in tasks requiring teamwork.

Multi-Agent Proximal Policy Optimization (MAPPO)

MAPPO extends Proximal Policy Optimization (PPO) to multi-agent settings using the Centralized Training with Decentralized Execution (CTDE) paradigm. During training, agents share information to develop better strategies but execute actions independently during gameplay.

Advantages

- Inter-Agent Coordination: Facilitates the development of cooperative or adversarial strategies among agents.
- Stable Training: PPO's inherent stability benefits are extended to multi-agent scenarios.

Limitations

- Computational Cost: Centralized training requires more computational resources.

Definition and Components

A Markov Decision Process (MDP) is defined as the tuple (S, A, dR , d0 , γ), where:

- S denotes the **state** space. For our environment:

  $$S \in R \text{^}(10x10) \backslash S`$$

  where S` represents the spaces occupied by obstacles or flag positions.

- A defines the **action** space:

  A = {Up (0, 1), Down (0, −1), Left (−1, 0), Right (1, 0), Stay (0, 0)}

Definition and Components

- **Rewards:** dR represents the reward distribution. For our problem:
  - **+100**: For capturing the opponent's flag.
  - **+25**: For successfully defending the flag.
  - **−25**: For getting caught while intruding.
  - **−2**: For staying in the same position.

Challenges

- Effectively shaping reward for exploration, defending, and capturing
- Achieving team objectives, when to start exploring to capture, when to defend the own territory.
- Delayed rewards for defending made it challenging for agents to learn.

Why MAPPO?

- **Centralized Training with Decentralized Execution:** Facilitates effective coordination among agents.
- **Proven Performance:** Achieves competitive or superior results in cooperative multi-agent scenarios.
- **Stable Learning Dynamics:** On-policy nature ensures stability in complex interactions.

Alignment with Problem Requirements

- **Coordination**: Enables agents to learn joint policies for balanced offensive and defensive strategies.
- **Stability:** Ensures stable learning in environments with complex agent interactions.

Model Parameters

Policy Network:

- **Input Layer:** 100 neurons (corresponding to the flattened observation space).
- **Hidden Layers:** Two fully connected layers with 128 neurons each, activated by ReLU functions.
- **Output Layer:** 5 neurons representing the action logits.
- **Optimizer:** Adam optimizer with a learning rate of 0.0003.
- **Policy Loss:** Clipped surrogate objective to ensure stability during training:

$$L_{\text{policy}} = -E\left[\min\left(r_t(\theta)A,\ \text{clip}\left(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon\right)A\right)\right]$$

where $r_t(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta\text{old}}(a|s)}$ is the probability ratio.

Metrics for Evaluation

- **High Win Rate:** The algorithm with a higher win rate indicates better team performance.
- **High Average Scores:** Reflects consistent ability to achieve objectives.
- **Low Draw Rate:** Indicates decisive outcomes, less stalemates.
- **Performance Stability:** Variance in scores across episodes to measure consistency.
- **Why These Metrics?** Assess overall dominance and effectiveness of each algorithm. Lower variance in scores suggests consistent performance.

Training Metrics: Rewards (IQL)

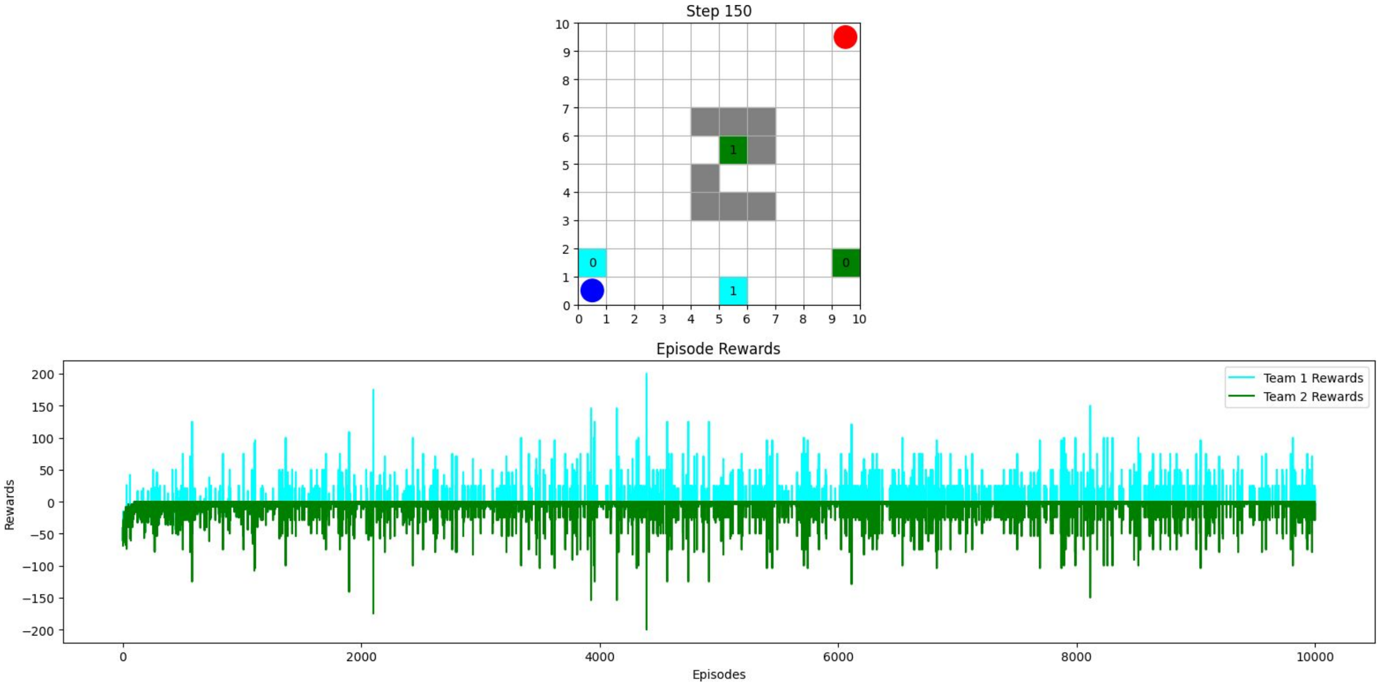


Figure: Loss progression during training (IQL)

Training Metrics: Loss and Rewards (MAPPO)

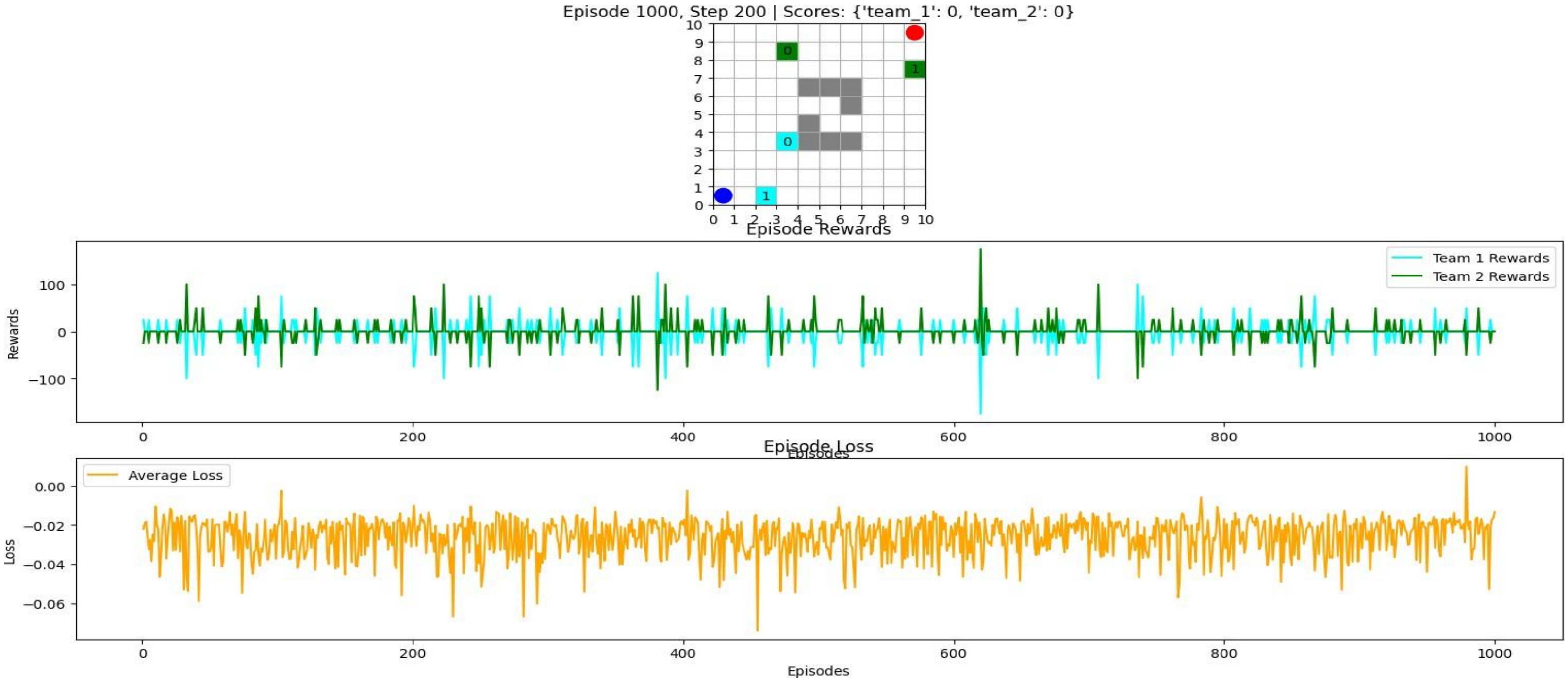
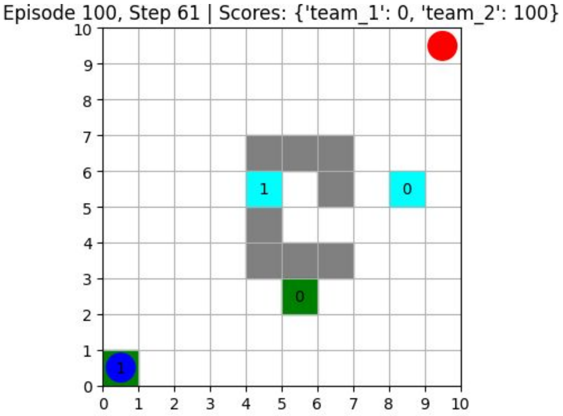
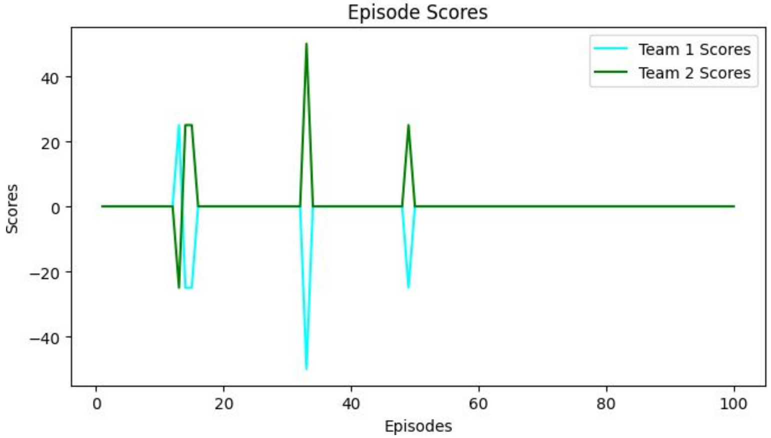


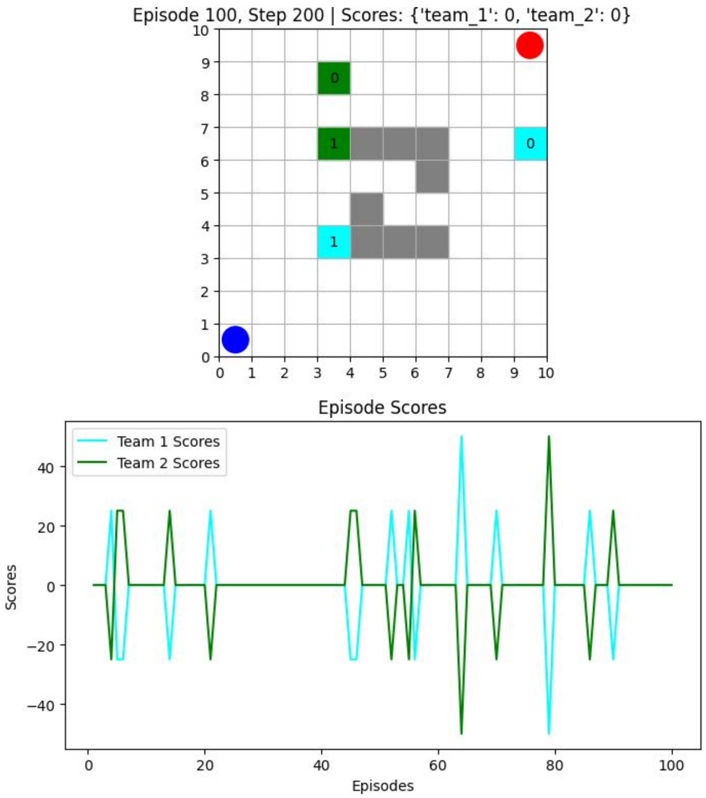
Figure: Loss progression during training (MAPPO)

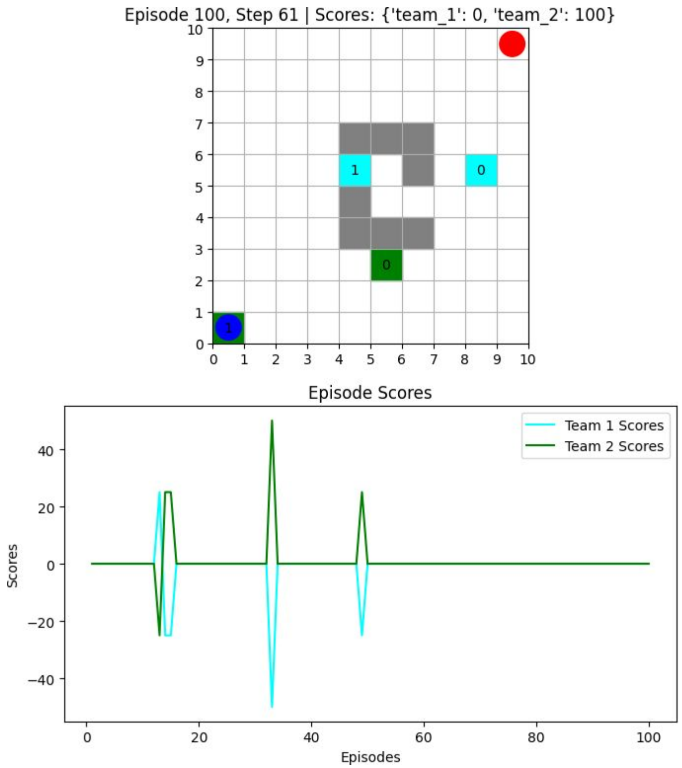
Testing Metrics: Rewards (IQL)





Testing Metrics: Loss and Rewards (MAPPO)





Results and Observations

Win and Draw Rates:

- MAPPO: Team 1 (53.13%), Team 2 (46.88%), Draws (0%).
- IQL: Team 1 (57.14%), Team 2 (42.86%), Draws (0%).

Average Scores:

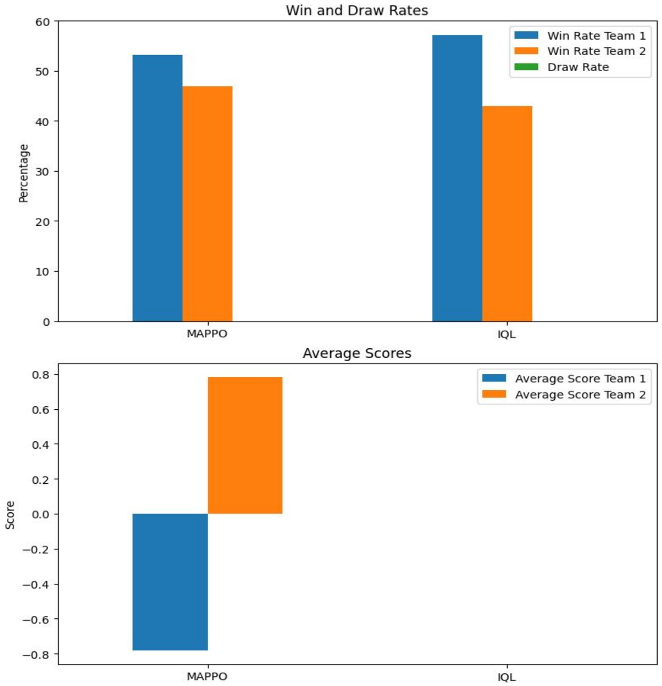
- MAPPO: Team 1 (-0.78125), Team 2 (0.78125).
- IQL: Team 1 (0.0), Team 2 (0.0).

Average Score Difference:

- MAPPO: -1.5625, indicating stronger dynamics between teams. IQL: 0.0, demonstrating balanced team performance.
- MAPPO reflects greater variability in team performance due to centralized policy training.

MAPPO have centralized policy training which reflects vairiability.

Results and Observations:



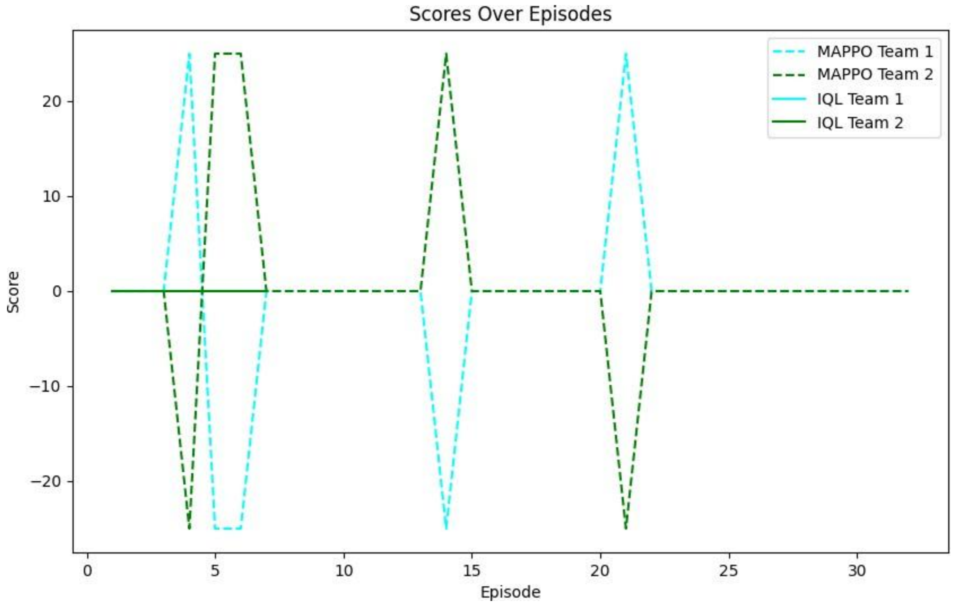


Figure: (Left) Win and Draw Rates; (Right) Scores Over Episodes.

Environment Comparison

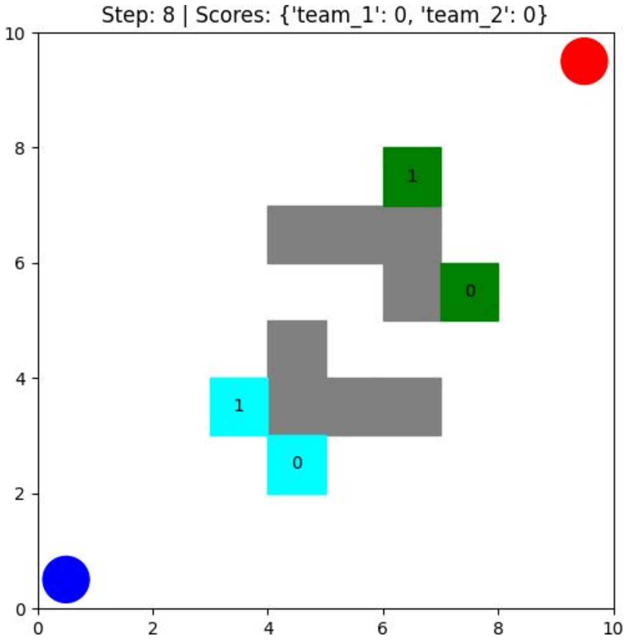


Figure: CTF Environment (Original)

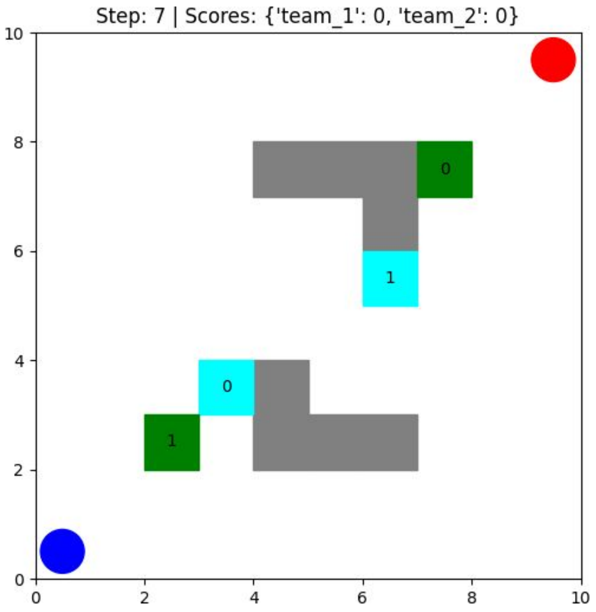


Figure: CTF Environment (Changed Obstacles)

Performance Metrics: Changed Environment

- **Win Rates:** MAPPO shows competitive balance (48.65% Team 1, 51.35% Team 2), while IQL favors Team 1 (53.33%).
- **Average Scores:** MAPPO results in balanced scores (2.03 for Team 1, -2.03 for Team 2), whereas IQL exhibits a significant score disparity (-5.00 for Team 1, 5.00 for Team 2).

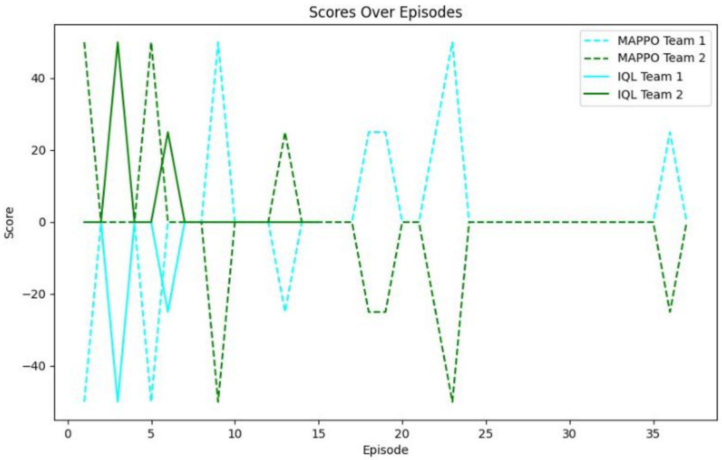
Inferences:

- MAPPO's centralized coordination leads to balanced gameplay.
- IQL's independent strategies result in unbalanced performance across teams.

Performance Metrics Over Episodes

Score Evolution:

- MAPPO exhibits stable dynamics with alternating scores over episodes, showing competitive engagement.
- IQL has steeper score fluctuations, indicating independent decisions often fail to adjust dynamically.



Score Dynamics Over Episodes

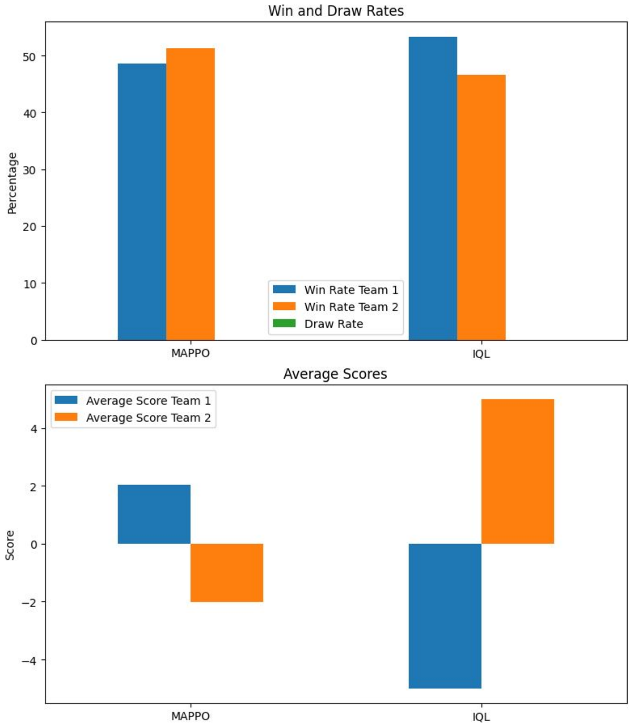
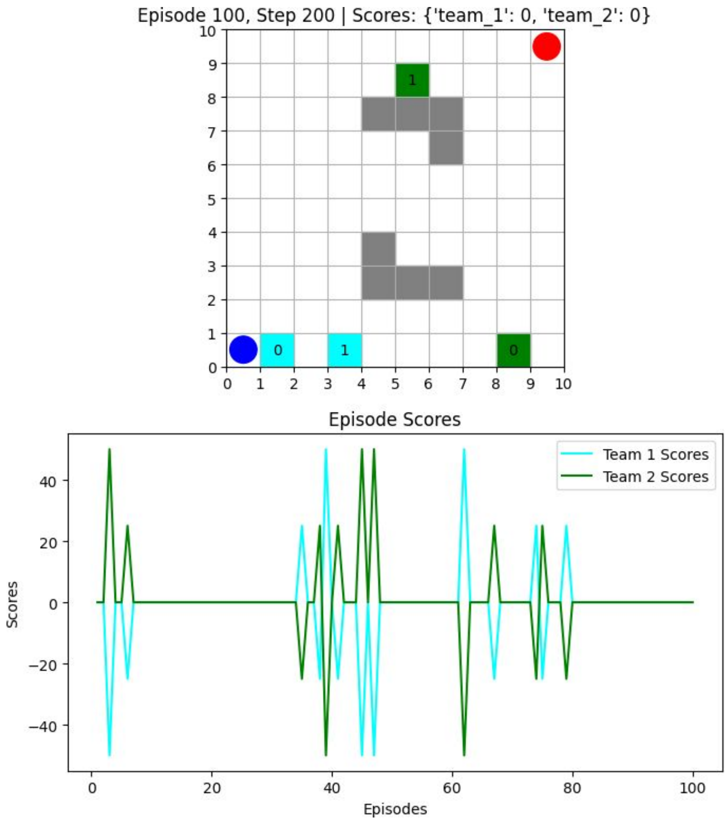
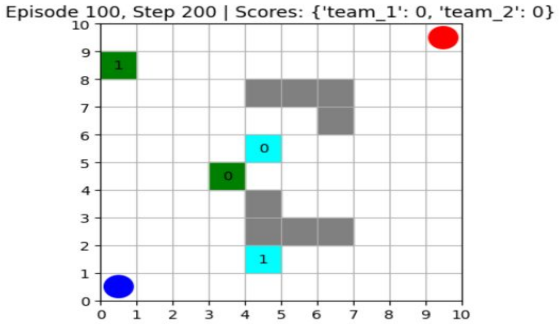


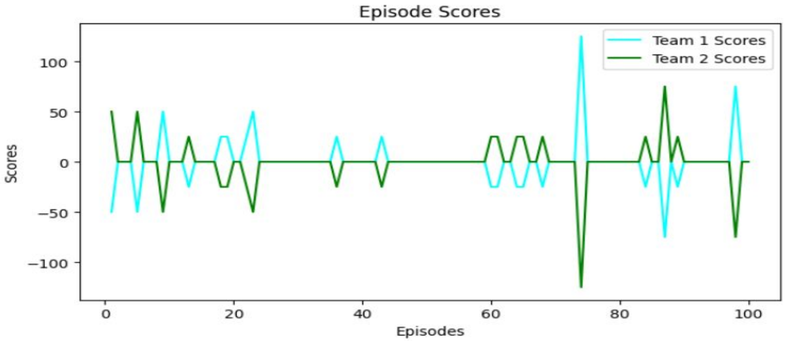
Figure: Win and Draw Rates

Testing Metrics: Rewards (IQL) in Changed Environment



Testing Metrics: Loss and Rewards (MAPPO) in Testing Environment





Conclusion & Key Findings:

MAPPO:

- Excels in environments requiring coordination and adaptability.
- Agents developed cooperative strategies, enhancing team performance.
- Performance remained stable across different environmental settings.

IQL:

- Effective in simpler environments with low coordination demands.
- Struggled with adaptability when the environment changed.
- Independent learning led to limitations in strategy development.

Final Thoughts

- This study underscores the importance of selecting appropriate MARL algorithms based on the specific requirements of the environment and tasks at hand.
- MAPPO's ability to foster coordination and adaptability makes it suitable for complex, dynamic scenarios like CTF.
- In contrast, IQL's simplicity makes it suitable for less demanding environments. Future work could explore hybrid approaches or alternative algorithms to further enhance multi-agent coordination and performance.

References

- **Sutton, R. S., & Barto, A. G.** (2018). *Reinforcement Learning: An Introduction*. MIT Press.
- **Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O.** (2017). *Proximal Policy Optimization Algorithms*. arXiv preprint arXiv:1707.06347.
- **Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I.** (2017). *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. Advances in Neural Information Processing Systems.
- **Busoniu, L., Babuska, R., & De Schutter, B.** (2008). *A Comprehensive Survey of Multi-Agent Reinforcement Learning*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews).