Paradigms of Machine Learning

Practical Activity 2: Multi-Agent Reinforcement Learning

David Piera i Jiménez
NIU: 1703730

Contents

1 Introduction **1**

2 Part 1: Prisoner's dilemma **1**
2.1 Independent Q-Learning (IQL) . . . 1
2.1.1 Implementation Details . . . 1
2.1.2 Training Dynamics Hyperparameter Analysis 2
2.1.3 Results 2
2.2 Centralized Q-Learning (CQL) . . . 3
2.2.1 Implementation Details . . . 3
2.2.2 Results 3
2.3 Conclusions 3

3 Part 2: Level-based Foraging **4**
3.1 Implementation and Methodology . . 4
3.2 Results 4
3.3 Conclusion 5

1 Introduction

This report presents the development and analysis of Multi-Agent Reinforcement Learning (MARL) systems applied to both game-theoretic and grid-world environments. The project is divided into two distinct phases:

- **Part 1** focuses on the *Prisoner's Dilemma*, a fundamental matrix game used to verify theoretical solution concepts such as the Nash Equilibrium and the Global Optimum.
- **Part 2** extends these algorithms to a complex, grid environment [**?**, **?**] requiring spatial coordination and navigation.

Through comparative analysis of training dynamics, convergence rates, and learned policies, this report evaluates the efficacy of decentralized versus centralized learning approaches in solving multi-agent coordination problems.

2 Part 1: Prisoner's dilemma

The objective of this section of the practical exercise is to explore and implement Multi-Agent Reinforcement Learning (MARL) techniques to solve the classic game theory problem known as the Prisoner's Dilemma. This exercise focuses on analyzing how different learning structures influence agent behavior in a competitive environment where individual rationality conflicts with collective benefit.

Specifically, this report compares two distinct approaches:

- **Independent Q-Learning (IQL):** A decentralized approach where two agents learn independently, treating each other as part of the environment.
- **Centralized Q-Learning (CQL):** A centralized approach where a single policy controls both agents, optimizing for the joint reward.

The primary goal is to observe the convergence behavior of these algorithms and identify the solution concepts they reach. We aim to determine whether independent agents inevitably converge to the Nash Equilibrium (Mutual Defection), which is individually rational but suboptimal, or if a centralized controller can achieve the Global Optimum (Mutual Cooperation) by maximizing the total welfare of the system.

2.1 Independent Q-Learning (IQL)

In this approach, we model the problem as a decentralized multi-agent system where two agents, A_1 and A_2, learn independently. Each agent maintains its own Q-table $Q_i(s, a_i)$ and treats the actions of the other agent as part of the environment's stochasticity. The goal is to observe if independent rational agents can learn to cooperate without explicit coordination mechanisms.

2.1.1 Implementation Details

The development of the IQL agent focused on implementing two critical methods within the provided IQL class structure. These functions define how the agent interacts with the environment and how it learns from its experiences.

Action Selection (act): To balance the trade-off between exploring new strategies and maximizing known rewards, I implemented an ϵ-greedy strategy. In this function, the agent se lects a random

action with probability ϵ (exploration), ensuring the state space is adequately traversed. Conversely, with probability $1-\epsilon$, the agent exploits its current knowledge by selecting the action associated with the highest value in its Q-table.

Q-Value Update (learn): This function is responsible for the core learning process. It implements the standard Q-learning update rule derived from the Bellman equation. Upon observing a transition, the agent updates the Q-value of the current state-action pair by shifting it towards a "target" value. This target is calculated as the sum of the immediate reward received and the discounted maximum Q-value of the next state. This mechanism allows the agent to iteratively refine its policy to maximize cumulative future rewards.

2.1.2 Training Dynamics Hyperparameter Analysis

To ensure the agents could effectively learn the optimal policy, a series of experiments were conducted to tune the hyperparameters. The primary focus was on the Learning Rate (α), which controls how quickly the agents update their Q-values based on new information.A grid search was performed with learning rates $\alpha \in \{0.01, 0.1, 0.5\}$. The training dynamics for each configuration revealed distinct behaviors:

Low Learning Rate ($\alpha = 0.01$): The learning process was stable but slow. The agents required a significant number of episodes to shift their value estimates from the initial values, delaying convergence.

Moderate Learning Rate ($\alpha = 0.1$): This proved to be the optimal configuration. It provided the best balance, allowing agents to learn quickly enough to adapt to the opponent's behavior while maintaining enough stability to converge to a steady Nash Equilibrium, as we can see in figure 1.

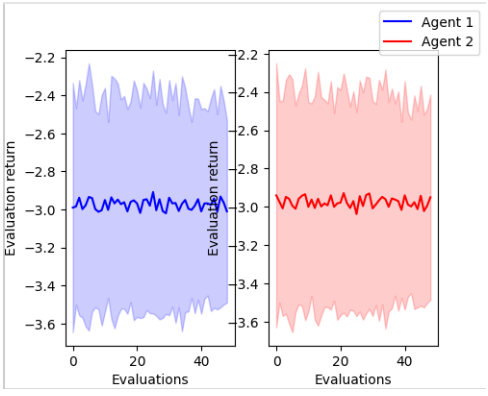

Figure 1: Evaluation LR = 0.1 (IQL)

High Learning Rate ($\alpha = 0.5$): This setting resulted in instability. The agents were highly sensitive to the stochastic exploration of their opponent, causing their Q-values and policy to oscillate rather than settle. As seen in the logs, this occasionally led to erratic returns (e.g., spikes dropping to -4.8) as agents failed to retain a consistent strategy, as we can see in the figure 2.

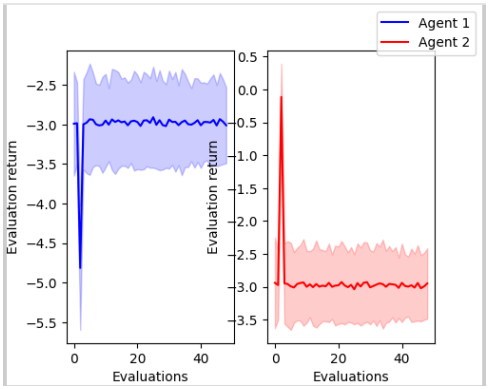

Figure 2: Evaluation LR = 0.5 (IQL)

Additionally, to optimize the exploration exploitation trade-off, the default linear decay for epsilon (ϵ) was replaced with an Exponential Decay schedule. This modification ensures that agents spend more time exploring in the early stages of training, preventing them from prematurely converging to a suboptimal policy before rapidly transitioning to exploitation to fine tune their behavior. Based on this empirical analysis, the final model was trained using $\alpha = 0.1$ and the exponential decay schedule.

2.1.3 Results

The training results obtained with the optimal configuration ($\alpha = 0.1$) demonstrate that independent agents fail to achieve the global optimum, instead settling into a suboptimal equilibrium. As observed in the performance evaluations (Figure 1), both agents converged to a stable mean return of approximately -3.0. In the context of the Prisoner's Dilemma payoff matrix, this specific value corresponds strictly to the outcome of Mutual Defection, indicating that neither agent learned to cooperate.

If we analyze the learned Q-values visualized in Figure 3. The plots reveal that the estimated return for "Defecting" stabilizes around -3.0, while the return for "Cooperating" settles near -5.0. This confirms that the Independent Q-Learning (IQL) algorithm has converged to the Nash Equilibrium, where the individually dominant strategy prevails over the collectively optimal one.

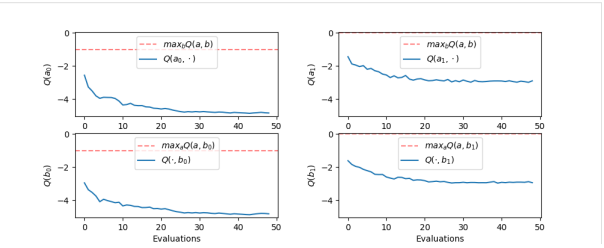

Figure 3: Q-Value Convergence Plot (IQL)

2.2 Centralized Q-Learning (CQL)

In contrast to the decentralized approach, we model the system as a centralized learner where a single global policy controls the joint actions of both agents, A_1 and A_2. This central controller maintains a unified Q-table that maps the full state of the environment and the combined actions of all agents to a single value. The goal is to determine if a centralized perspective, which optimizes for the collective return rather than individual gain, can identify and converge to the global optimum (Mutual Cooperation).

2.2.1 Implementation Details

The implementation of the Centralized Q-Learning agent required a significant shift in how actions are handled. Instead of having two separate agents making independent decisions, I created a single central controller that chooses a "joint action" for the entire team at once. This means the agent doesn't just pick "Cooperate" or "Defect" for one person; it picks a combined move for both (such as "Both Cooperate" or "Agent 1 Defects, Agent 2 Cooperates").

The most important change, however, was in how the agent learns from rewards. In the previous independent approach, each agent only cared about its own personal score. For this centralized version, I modified the learning rule to calculate the sum of all rewards ($R_{total} = r_1 + r_2$). This simple change forces the agent to look at the team score rather than individual scores. By maximizing the group's total points, the algorithm naturally stops seeing betrayal as a good strategy and instead learns to prefer the option where both agents win together.

I directly adopted the optimal configuration identified during the IQL phase: a learning rate of $\alpha = 0.1$ and the exponential decay schedule for exploration. The training dynamics proved to be remarkably stable under these settings

2.2.2 Results

The results obtained from the Centralized Q-Learning agent mark a decisive shift from the outcomes observed in the independent setting.

As illustrated in Figure 4, the performance profile is radically different from the IQL agents. Instead of converging to -3.0, both agents stabilize at a mean return of approximately -1.0 per episode. In the context of the Prisoner's Dilemma, a return of -1 for both agents corresponds strictly to Mutual Cooperation. This indicates that the centralized policy has successfully located and stabilized at the Global Optimum.

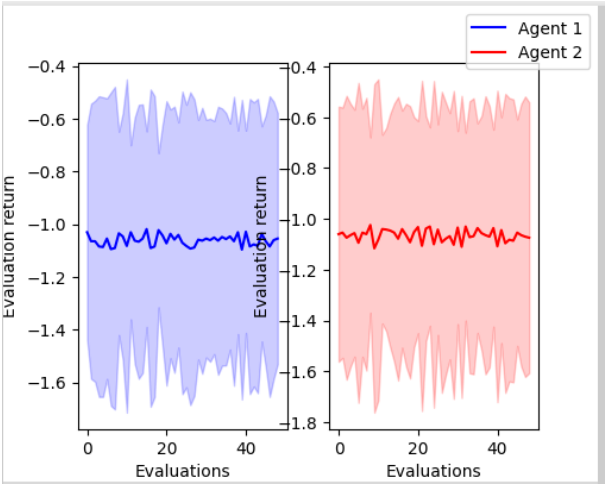

Figure 4: Evaluation (CQL)

Looking at Figure 5, we can see the agent learns that the expected sum of rewards for Mutual Cooperation is -2.0 (Green line), whereas the sum for Mutual Defection is -6.0 (Red line). Because the central agent optimizes for the total group reward, it strictly prefers the action with the higher value ($-2 > -6$). Unlike the independent agents, which were trapped by individual incentives, the central agent effectively identifies the cost of defection, making cooperation the dominant strategy.

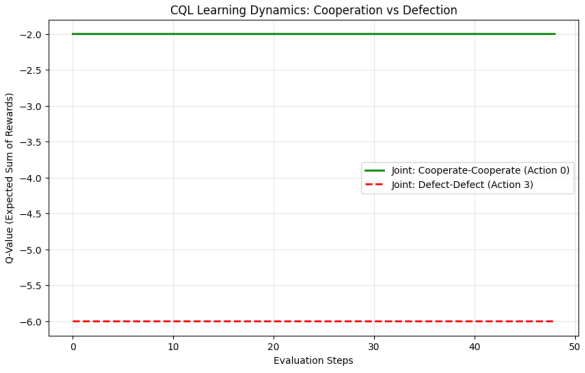

Figure 5: Convergence Plot (CQL)

2.3 Conclusions

The comparative analysis of Independent Q-Learning (IQL) and Centralized Q-Learning (CQL) on the

Prisoner's Dilemma demonstrates the critical impact of learning architecture on multi-agent behavior.

The Independent Q-Learning agents, acting as decentralized rational entities, inevitably converged to the Nash Equilibrium of Mutual Defection (Return: -3.0). Despite the existence of a better collective outcome, the lack of coordination and the risk of exploitation forced each agent to prioritize individual gain, trapping the system in a suboptimal state.

In contrast, the Centralized Q-Learning agent successfully achieved the Global Optimum of Mutual Cooperation (Return: -1.0). By transforming the problem into a single-objective optimization task using the sum of rewards, the central controller eliminated the conflict of interest inherent in the dilemma.

3 Part 2: Level-based Foraging

The second phase of this project extends the study of Multi-Agent Reinforcement Learning to the Level-Based Foraging environment. Unlike the static Prisoner's Dilemma, LBF is a grid-world environment that introduces spatial complexity and navigation tasks. In this scenario, agents are assigned specific "levels" and must navigate a 5x5 grid to collect food items, which also possess a level.

The critical challenge in this environment is the coordination requirement: an item can only be collected if the sum of the levels of the participating agents equals or exceeds the food's level. This creates dynamic interdependencies where agents must decide whether to forage alone (if their level suffices) or coordinate with partners to secure larger rewards. This section compares the efficacy of Independent Q-Learning (IQL) and Centralized Q-Learning (CQL) across two distinct configurations: a standard mode where agents are inherently competitive regarding credit assignment, and a cooperative mode where rewards are shared, enforcing a unified objective.

3.1 Implementation and Methodology

We took the Q-learning code from Part 1 and upgraded it to work in the Level-Based Foraging (grid world) game. This game is much harder than the previous one because the agents have to move around a map rather than just picking a single option. We run the training for 100,000 rounds (episodes) to give them enough time to learn.

Hyperparameter Adaptation (Settings) Because the grid map is much bigger and more complex than the simple Prisoner's Dilemma, the agents need more time to explore. We adjusted the "exploration rate" (ϵ) so that the agents spend the first 90,000 steps (90% of the time) acting somewhat randomly. This prevents them from getting stuck in bad habits early on and ensures they find food sources before settling on a final strategy.

In the Independent Q-Learning (IQL) configuration, the system operates as two completely separate players who cannot communicate. Each agent only sees its immediate surroundings, converting that limited view into a simple code, and maintains a private scorecard (Q-table) to track its progress. Crucially, they learn entirely based on their own personal points; they have no awareness of the other agent's plans or the global map, effectively treating their partner as just another unpredictable part of the environment.

In contrast, the Centralized Q-Learning (CQL) approach controls both agents simultaneously. Instead of limited individual views, this central controller sees the entire picture combining what both agents see into a joint state and selects "team moves" where both act together. Because it learns based on the team's total score rather than individual points, it naturally forces cooperation; selfish actions that don't benefit the group result in zero reward, teaching the system that winning requires working together.

3.2 Results

The experimental evaluation of the Independent Q-Learning (IQL) and Centralized Q-Learning (CQL) algorithms in the Level-Based Foraging environment demonstrates distinct learning dynamics, particularly when comparing standard (competitive) and cooperative modes. The following analysis is based on the moving average of the total team reward collected over 100,000 training episodes.

In the standard environment configuration (coop=False), agents are individually rewarded for their contribution to collecting food. As illustrated in the Figure below, both IQL (Blue) and CQL (Red) successfully learn to solve the task, converging to a near optimal total team reward of approximately 1.0 . This indicates that even in a decentralized setting, agents can learn to navigate effectively to food sources when individual rewards are directly aligned with the task. However, the Centralized Q-Learning agent demonstrates a slightly faster learning curve, reaching high reward levels earlier (around episode 40,000) compared to the Independent agent, which stabilizes closer to episode 50,000.

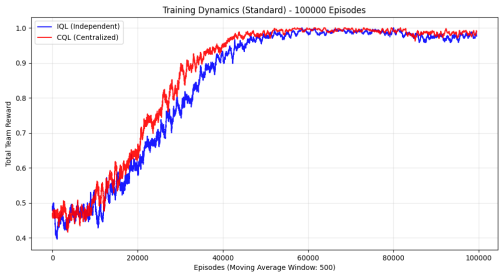

Figure 6: Results Standard

When the environment is switched to the cooperative configuration (coop=True), the reward structure is shared, meaning both agents receive the total reward regardless of individual contribution. Figure 2 reveals that the learning curves for this mode are notably steeper than in the standard mode, with agents beginning to acquire significant rewards as early as episode 10,000. The Centralized Q-Learning agent again outperforms the Independent approach during the initial learning phase; by approximately episode 30,000, CQL has nearly maximized the reward, whereas IQL requires approximately 10,000 additional episodes to reach the same level of performance. The shared reward signal appears to simplify the learning problem by aligning the agents' incentives perfectly, preventing greedy behaviors that might arise in competitive settings.

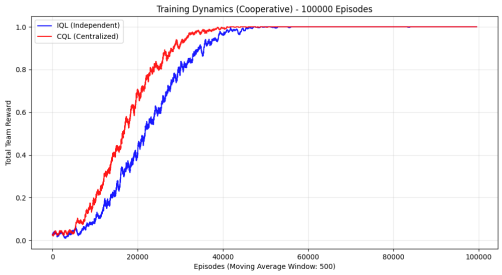

Figure 7: Results cooperative

To further validate these quantitative results, video recordings of the trained agents were generated for both algorithms in both modes. These visualizations confirm the data shown in the plots, revealing that the trained agents exhibit direct pathing toward food sources, minimizing unnecessary steps. In the cooperative videos specifically, agents demonstrate synchronized movement, often arriving at food locations simultaneously.

3.3 Conclusion

This study extended the application of Multi-Agent Reinforcement Learning from simple matrix games to a complex, high-dimensional grid world, supporting several key conclusions regarding the efficacy of decentralized versus centralized learning. Across both standard and cooperative scenarios, Centralized Q-Learning (CQL) consistently demonstrated faster convergence rates than Independent Q-Learning (IQL). By treating the multi-agent system as a single optimization problem, CQL bypasses the non-stationarity issues that plague independent learners, where one agent's learning constantly changes the environment for the other.

Despite being slower, IQL proved to be a robust approach, eventually converging to the same optimal performance as CQL. This suggests that for this easy task, learners can still achieve global optimality given sufficient training time. Furthermore, the transition from standard to cooperative rewards significantly improved the learning speed for both algorithms. In summary, while Centralized Q-Learning offers superior sample efficiency and stability, Independent Q-Learning remains a viable and scalable alternative for environments where full centralization is computationally difficult.