Decentralized learning in Overcooked

Qin Ariel Xu
School of Information Science
Mcgill University
qin.xue@mail.mcgill.ca

Abstract

This project investigates decentralized multi-agent reinforcement learning (MARL) in the Overcooked-AI environment, where agents must coordinate to complete cooperative cooking tasks. Four reinforcement learning algorithms were implemented from scratch, including Q-learning, DQN, PPO, and TRPO. Their performance was evaluated under both homogeneous (two RL agents) and heterogeneous (RL + heuristic agent) settings. Experiments focused on the emergence of coordination, learning dynamics, and the impact of partner reliability across different kitchen layouts.

1 Introduction

Multi-agent reinforcement learning (MARL) addresses coordination challenges among agents working toward shared goals. This project explores cooperative MARL in the Overcooked-AI environment, where agents must jointly prepare and deliver meals—requiring complex coordination in task assignment, movement, and timing. The study compares value-based and policy-gradient RL algorithms to analyze the emergence of cooperative behaviors under decentralized learning. Algorithms are implemented from scratch and evaluated on two kitchen layouts of varying complexity. The project offers insights into performance, adaptability, and coordination in decentralized cooperative RL.

2 Background

The Overcooked-AI environment is modeled as a two-player Markov game $(S, A_1, A_2, P, R, \gamma)$, where agents select actions simultaneously and observe joint transitions. Unlike single-agent MDPs, the decentralized setting features agents with individual policies and no access to each other's internal strategies—only shared state and team rewards. The project examine two setups: heterogeneous (RL agent + random partner), where the environment becomes non-stationary due to the partner's unpredictable behavior, violating standard MDP assumptions [6]; and homogeneous (two RL agents), where mutual policy adaptation poses coordination challenges typical in decentralized MARL [2]. Despite no explicit communication, full observability enables agents to infer partner behavior from the shared state.

3 Related work

Cooperative multi-agent reinforcement learning (MARL) has emerged as a key area in artificial intelligence, addressing scenarios where multiple agents must coordinate to achieve shared goals. Cooperative MARL faces several fundamental challenges, including non-stationarity, coordination, and credit assignment [3]. To address these, the centralized training with decentralized execution (CTDE) framework has gained popularity. Methods such as MADDPG [6] and QMIX [8] exemplify this approach, enabling agents to learn cooperative strategies centrally while acting independently at test time.

While CTDE methods have shown success, fully decentralized MARL—where agents learn independently without shared gradients or direct communication—remains particularly challenging. Early work demonstrated that independent Q-learning can converge to suboptimal policies due to agents ignoring each other's learning dynamics [7]. Despite this, decentralized settings are essential for real-world applications with limited communication or centralized infrastructure. Recent advances, including value decomposition techniques[8, 9], attempt to balance local decision-making with global coordination objectives.

This project builds on these foundational insights by using Overcooked-AI to evaluate and compare decentralized cooperative learning algorithms. It focuses on the ability of agents to coordinate effectively under varying conditions, particularly when paired with non-stationary or unpredictable partners. In doing so, it contributes to a deeper understanding of algorithmic robustness, generalization, and the challenges inherent in decentralized coordination.

4 Method

This project is built on a lightweight, open-source Overcooked Gym wrapper that replicates the cooperative cooking task from the original Overcooked-AI environment[10]. It follows the OpenAI Gym API and supports multiple kitchen layouts, making it suitable for custom reinforcement learning (RL) pipelines. The environment is grid-based and includes interactive objects such as onion dispensers (O), soup pots (P), dish stacks (D), and delivery stations (S).

Two agents operate concurrently in the kitchen, each with five discrete actions: `up`, `down`, `left`, `right`, and `act`, where `act` performs context-dependent operations (e.g., picking up or placing items). The environment computes joint transitions and assigns a shared team reward based on both agents' actions. The reward function is primarily sparse: agents receive +20 for each delivered soup, with optional shaping rewards (+2 for adding an onion, +5 for picking up soup) and a time penalty of -0.1 per step, leading to a maximum penalty of -40 per episode. Each episode is capped at 400 steps.

Four RL algorithms—Q-learning, Deep Q-Network (DQN), Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO)—were implemented from scratch. Each was tested in two coordination settings. In the heterogeneous setting, a single RL agent is paired with a heuristic partner that selects random actions, introducing high unpredictability and forcing the agent to adapt to a non-stationary teammate. In the homogeneous setting, both agents are controlled by the same RL algorithm but learn independently with separate networks, modeling a decentralized learning scenario without centralized coordination or parameter sharing.

All agents operate under full observability, with access to the complete environment state, including the positions, orientations, and hand contents of both agents, as well as the status of relevant objects. In the homogeneous setup, agents learn simultaneously by independently selecting actions and updating their policies based on a shared team reward. To establish a baseline understanding of the challenges and inherent capabilities of each algorithm, no coordination mechanisms, reward shaping, or auxiliary training techniques were employed. Any emergence of coordinated behavior is therefore attributed solely to the shared environment state and the common reward signal.

This structured setup has varying coordination conditions, algorithmic diversity, and realistic environmental constraints, which enables a comprehensive evaluation of how different RL methods perform in cooperative multi-agent tasks with differing levels of partner reliability and interaction complexity.

5 Experiments

The experiments in this project are designed to evaluate the performance of various reinforcement learning (RL) algorithms in a cooperative multi-agent task using the Overcooked-AI environment. A consistent experimental protocol was followed, encompassing controlled layout settings, systematic coordination modes, algorithm-specific hyperparameter tuning, and full observability for all agents.

5.1 Training Protocol

All experiments were conducted with a fixed episode length of 400 time steps to ensure consistent comparability between layouts and algorithms. Each configuration involved two sequential phases:

1. **Hyperparameter Tuning Phase (1,000 episodes):** Each algorithm was tuned over a range of hyperparameters. The best-performing configuration was selected based on average episode return during the final portion of training.
2. **Final Evaluation Phase (5,000 episodes):** The optimal configuration was subsequently trained for 5,000 episodes to evaluate long-term learning dynamics and coordination emergence. In cases where the agent demonstrated promising learning trends within 5,000 episodes or had shown effective performance in the simpler layout, extended training beyond 5,000 episodes was conducted to further assess its convergence potential.

To conserve resources and maintain focus on coordination behavior rather than tuning dynamics, intermediate results from the tuning phase were not exhaustively saved.

5.2 Environment Layouts

Two layouts from the Overcooked-AI Gym wrapper were selected to represent varying coordination challenges:

- **Simple Layout (simple_l):** Features minimal task conflict, offering a low-difficulty baseline for evaluating coordination.
- **Complex Layout (simple_kitchen):** Includes bottlenecks, constrained movement areas, and structural interferences, increasing task difficulty and requiring implicit planning and avoidance strategies.

5.3 Algorithmic Hyperparameter Tuning

Each RL algorithm was tuned during the 1,000-episode phase using the following ranges:

- **Q-learning:** $\alpha \in [0.05, 0.2]$, $\epsilon \in [0.1, 0.5]$, $\gamma = 0.99$.
- **DQN:** $\epsilon \in [0.1, 0.5]$ (with/without decay), $\gamma = 0.99$, batch size $\in [32, 128]$.
- **PPO:** $\gamma \in [0.95, 0.99]$, learning rate $\in [10^{-4}, 10^{-2}]$, clip parameter $\in [0.05, 0.2]$, epochs $K \in [4, 8]$, entropy coefficient $\in [0.01, 0.1]$.
- **TRPO:** $\gamma \in [0.99, 0.999]$, max KL $\in [10^{-4}, 10^{-2}]$, CG iterations $\in [5, 15]$, damping $\in [10^{-4}, 10^{-2}]$, line search steps $\in [5, 15]$, policy update epochs $\in [5, 10]$.

5.4 Evaluation

Agent performance was evaluated using episode-level cumulative rewards, with particular attention to learning dynamics, convergence behavior, and adaptation to partner strategies. The analysis emphasized the emergence of coordination rather than statistical significance testing. As the environment does not include a `Done` signal, the final episode reward serves as the primary indicator of learning efficiency and effectiveness—higher cumulative rewards indicate better overall performance.

6 Comparative Results and Explanations

(A) Heterogeneous Setting (RL Agent + Heuristic Partner)

Q-learning + Heuristic simple_l & simple_kitchen: Both layouts resulted in no meaningful learning progress. The persistent lack of improvement has shown the inability of Q-learning to handle environments characterized by unpredictable teammates. The simplicity or complexity of the layout has negligible impact.

DQN + Heuristic simple_l & simple_kitchen: In both layouts, DQN initially exhibits minor adaptive trends but eventually stagnates or regresses. Although the `simple_kitchen` scenario shows a slightly more pronounced initial learning trend, both ultimately fail similarly.

PPO + Heuristic simple_l & simple_kitchen: PPO agents consistently fail to achieve meaningful learning in both scenarios. The added complexity in `simple_kitchen` has little additional impact. PPO remains ineffective under sparse and delayed rewards, as seen clearly in both layouts.

TRPO + Heuristic simple_l & simple_kitchen: Both layouts show TRPO briefly achieving minor improvements early on but rapidly deteriorating to minimal reward levels.

Given that the presence of a random heuristic partner drastically reduced learning effectiveness for all algorithms, the non-stationary behavior caused by stochastic teammates may complicate learning and prevent convergence to optimal or even viable coordination strategies.

(B) Homogeneous Setting (Two RL Agents)

Two-agent Q-learning **simple_l:** Rapid convergence, stable at around 250 rewards within 1000 episodes. **simple_kitchen:** Successful but significantly slower learning, taking roughly 10,000–20,000 episodes to reach a comparable stable reward of around 250.

The contrast in convergence speed between layouts demonstrates the profound impact environment complexity has on tabular Q-learning. While the simpler layout allows quick discovery and exploitation of effective strategies, the complexity of `simple_kitchen` demands significantly prolonged exploration and adaptation phases before convergence. Nevertheless, the eventual stable performance highlights Q-learning's potential when sufficient computational resources (time and exploration episodes) are provided.

Two-agent DQN **simple_l:** Demonstrated clear phases of initial stagnation, rapid improvement, and subsequent stabilization at a high reward level.
simple_kitchen: Severely impeded learning; minimal improvements observed only after extended training, and no stable convergence was achieved.

The striking difference observed for DQN between `simple_l` and `simple_kitchen` underscores the high sensitivity of neural value approximation methods to environmental complexity and non-stationary agent interactions. The pronounced difficulty and eventual failure to converge in `simple_kitchen` may be due to DQN's struggles with effective generalization and stable value estimation in dynamically complex multi-agent scenarios.

Two-agent PPO and TRPO Both PPO and TRPO displayed no meaningful learning progress in either layout. Their conservative and constrained update mechanisms, which theoretically stabilize single-agent learning, severely limit the exploration necessary for emergent cooperative strategies under decentralized training. The increased environmental complexity in `simple_kitchen` offers no additional insights due to fundamental algorithmic incompatibilities in these MARL contexts.

7 Discussion and Conclusion

This study explored the performance of Q-learning, DQN, PPO, and TRPO in decentralized cooperative multi-agent settings across both homogeneous (two RL agents) and heterogeneous (RL agent with a random partner) configurations. The findings emphasize some key challenges.

A major challenge is the non-stationarity introduced when agents learn simultaneously. As each agent updates its policy, the environment dynamics shift from the perspective of others, violating the Markov assumption and impeding stable learning [3]. Sparse and delayed rewards further complicate learning by limiting feedback, making it difficult for agents to associate actions with outcomes—especially in tasks requiring precise joint execution[5].

Value-based methods (Q-learning, DQN) showed better performance in simpler layouts, where the state and action spaces were smaller and coordination demands lower. However, these methods struggled in more complex environments due to challenges in stable value estimation under sparse feedback and dynamic teammates[2]. Policy-gradient methods, such as PPO and TRPO, offer greater flexibility and support for stochastic policies. However, their poor performance in the decentralized setting contrasts with their success in centralized training scenarios [11]. This discrepancy may be due to the absence of shared information or the lack of an explicit coordination mechanism in the decentralized framework.

Notably, all algorithms failed to learn effective policies in the heterogeneous setting when paired with a randomly acting partner. This outcome is likely due to the high level of non-stationarity introduced by the unpredictable teammate, which hindered the agent's ability to develop a stable and consistent coordination strategy.

To address these limitations, future work should explore centralized training with decentralized execution (CTDE)[6], value decomposition approaches like QMIX [8], and exploration strategies based on intrinsic motivation or reward shaping [4]. Experience-sharing mechanisms may also enhance learning speed and sample efficiency in cooperative environments [1].

In conclusion, cooperative MARL remains a difficult domain for standard RL methods. The combination of non-stationarity, sparse rewards, and the need for implicit coordination presents significant obstacles. While value-based and policy-gradient methods each offer strengths, neither is sufficient on its own. Advancing MARL requires dedicated techniques that support stable coordination under uncertainty and dynamic interaction patterns.

References

[1] Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. *Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning*. May 19, 2021. DOI: 10.48550/arXiv.2006.07169. arXiv: 2006.07169[cs]. URL: http://arxiv.org/abs/2006.07169 (visited on 04/13/2025).

[2] Jakob Foerster et al. *Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning*. May 21, 2018. DOI: 10.48550/arXiv.1702.08887. arXiv: 1702.08887[cs]. URL: http://arxiv.org/abs/1702.08887 (visited on 04/13/2025).

[3] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. "A Survey and Critique of Multiagent Deep Reinforcement Learning". In: *Autonomous Agents and Multi-Agent Systems* 33.6 (Nov. 2019), pp. 750–797. ISSN: 1387-2532, 1573-7454. DOI: 10.1007/s10458-019-09421-1. arXiv: 1810.05587[cs]. URL: http://arxiv.org/abs/1810.05587 (visited on 04/10/2025).

[4] Shariq Iqbal and Fei Sha. *Coordinated Exploration via Intrinsic Rewards for Multi-Agent Reinforcement Learning*. May 22, 2021. DOI: 10.48550/arXiv.1905.12127. arXiv: 1905.12127[cs]. URL: http://arxiv.org/abs/1905.12127 (visited on 04/13/2025).

[5] Boyin Liu et al. "Lazy Agents: A New Perspective on Solving Sparse Reward Problem in Multi-agent Reinforcement Learning". In: *Proceedings of the 40th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, July 3, 2023, pp. 21937–21950. URL: https://proceedings.mlr.press/v202/liu23ac.html (visited on 04/13/2025).

[6] Ryan Lowe et al. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. Mar. 14, 2020. DOI: 10.48550/arXiv.1706.02275. arXiv: 1706.02275[cs]. URL: http://arxiv.org/abs/1706.02275 (visited on 04/10/2025).

[7] Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. "Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems". In: *The Knowledge Engineering Review* 27.1 (Feb. 2012), pp. 1–31. ISSN: 1469-8005, 0269-8889. DOI: 10.1017/S0269888912000057. URL: https://www.cambridge.org/core/journals/knowledge-engineering-review/article/abs/independent-reinforcement-learners-in-cooperative-markov-games-a-survey-regarding-coordination-problems/05C8A9CB66528DA19B855B8C5DFBA981 (visited on 04/14/2025).

[8] Tabish Rashid et al. *QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*. June 6, 2018. DOI: 10.48550/arXiv.1803.11485. arXiv: 1803.11485[cs]. URL: http://arxiv.org/abs/1803.11485 (visited on 04/10/2025).

[9] Kyunghwan Son et al. *QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning*. May 14, 2019. DOI: 10.48550/arXiv.1905.05408. arXiv: 1905.05408[cs]. URL: http://arxiv.org/abs/1905.05408 (visited on 04/10/2025).

[10] Inês Vieira. *ilvieira/overcooked-gym*. original-date: 2022-10-24T22:40:32Z. Oct. 24, 2022. URL: https://github.com/ilvieira/overcooked-gym (visited on 04/10/2025).

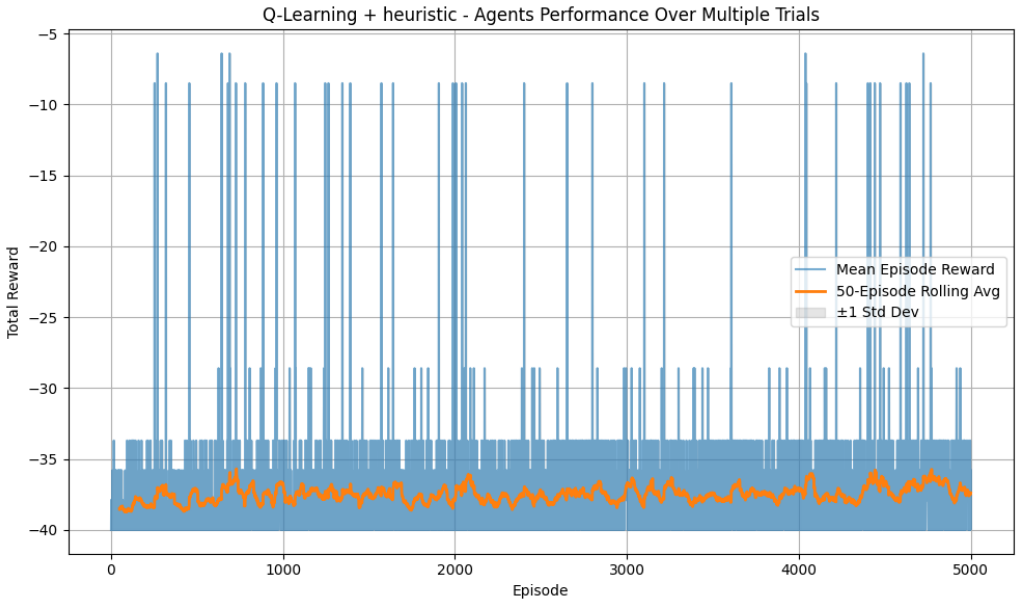
[11] Chao Yu et al. *The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games*. Nov. 4, 2022. DOI: 10.48550/arXiv.2103.01955. arXiv: 2103.01955[cs]. URL: http://arxiv.org/abs/2103.01955 (visited on 04/13/2025).

Appendix

Evaluation graphs (more graphs for tuning can be found in notebook)

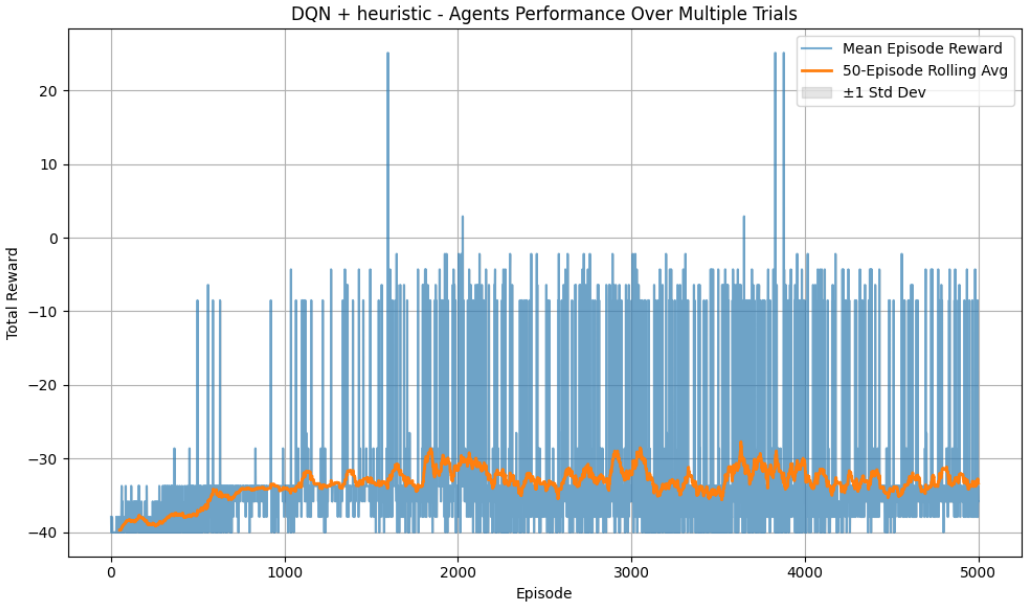
Simple layout - Q-Learning + heuristic

Evaluation: alpha=0.1, gamma=0.99, epsilon=0.1



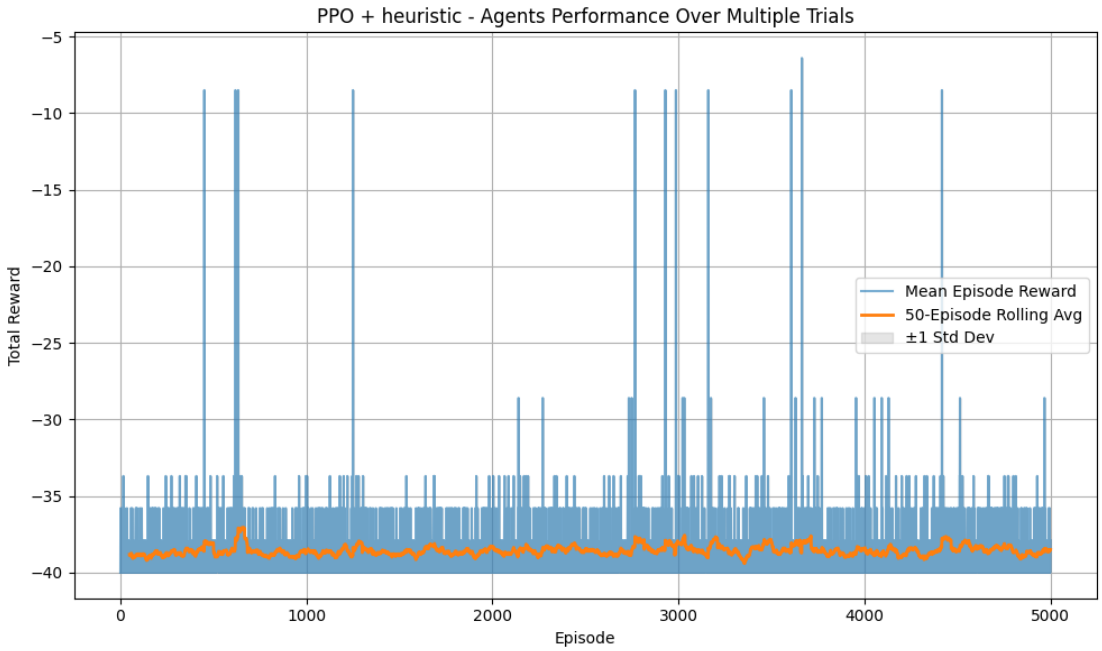
Simple layout - DQN + heuristic

Evaluation: gamma = 0.99, epsilon = 0.1, epsilon_decay = 1, min_epsilon = 0.1,batch_size = 128, update_target_every = 20, lr = 1e-5



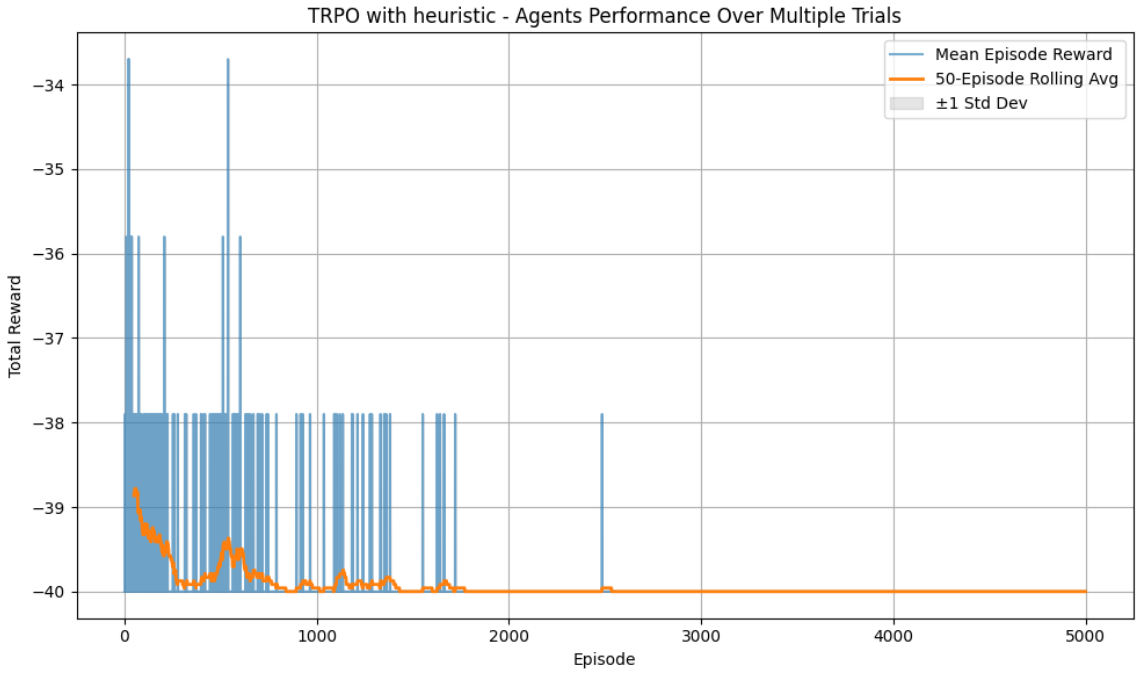
Simple layout - PPO + heuristic

Evaluation: gamma=0.95, lr=1e-4, eps_clip=0.05, K_epochs=6, entropy_coef = 0.01



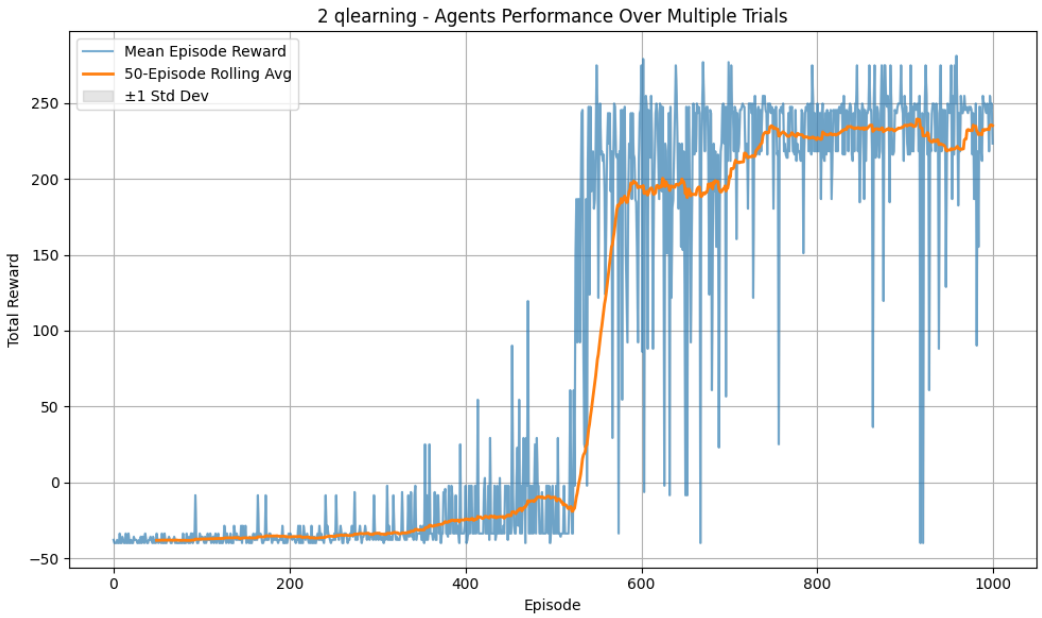
Simple layout - TRPO + heuristic

Evaluation: gamma=0.99, max_kl=1e-2, cg_iters=10, damping=1e-2, ls_max_steps=10, ls_backtrack_coeff=0.8, lr=1e-4, update_epochs=10



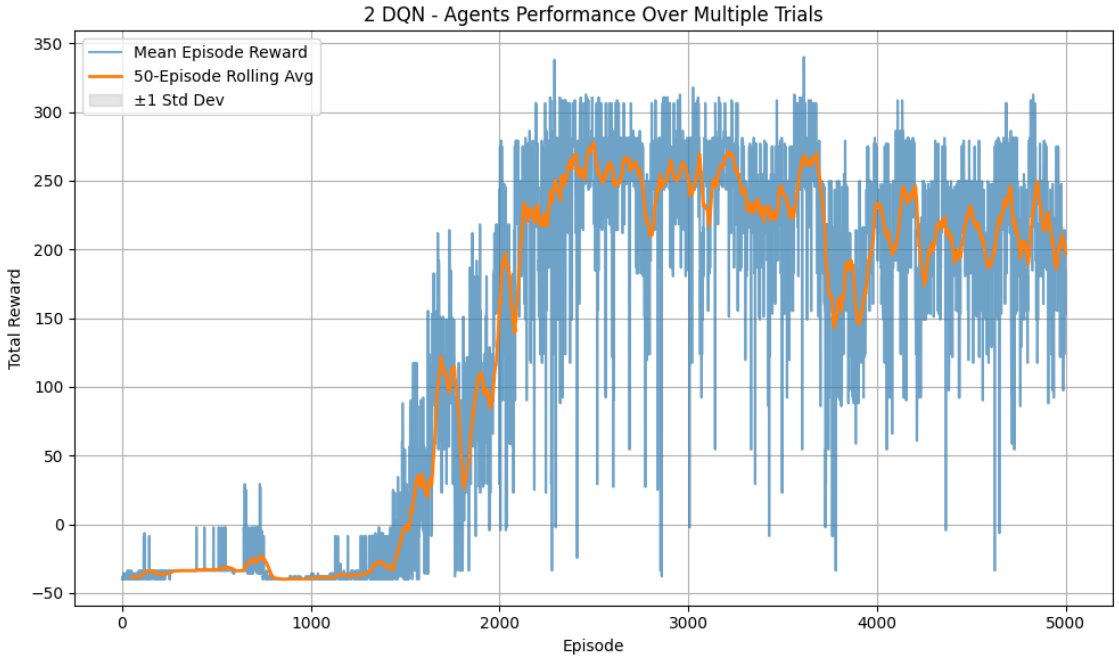
Simple layout – 2 Q learning

Evaluation: alpha=0.1, gamma=0.99, epsilon=0.1



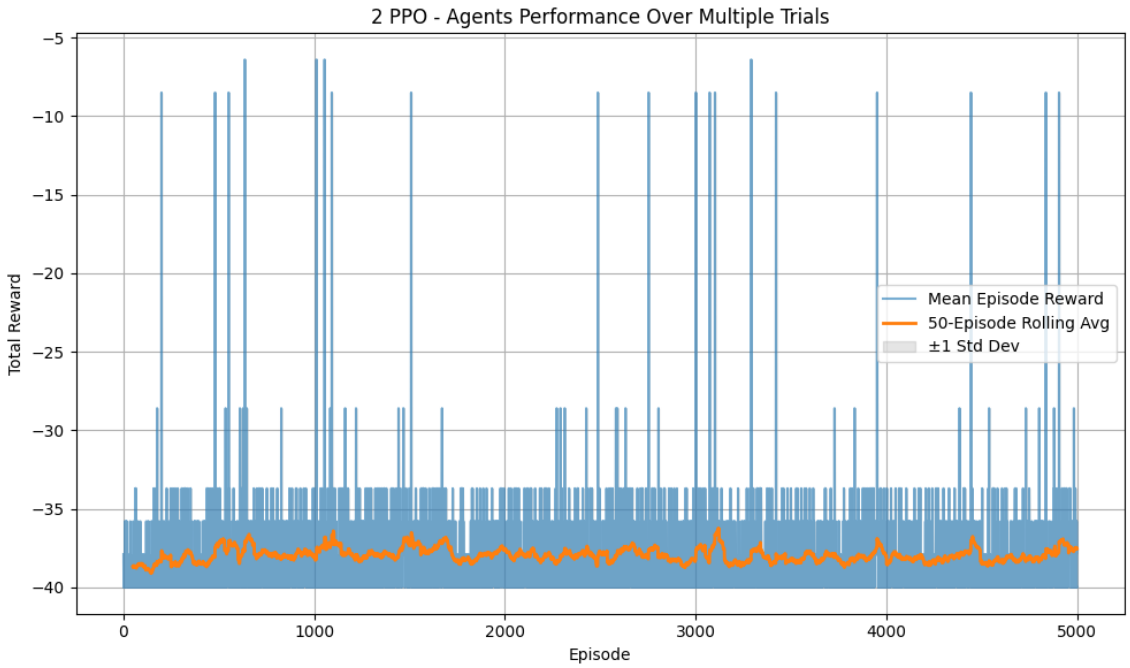
Simple layout – 2 DQN

Evaluation: gamma = 0.99, epsilon = 1.0, epsilon_decay = 0.995, min_epsilon = 0.2, batch_size =128, update_target_every = 20, lr=1e-4



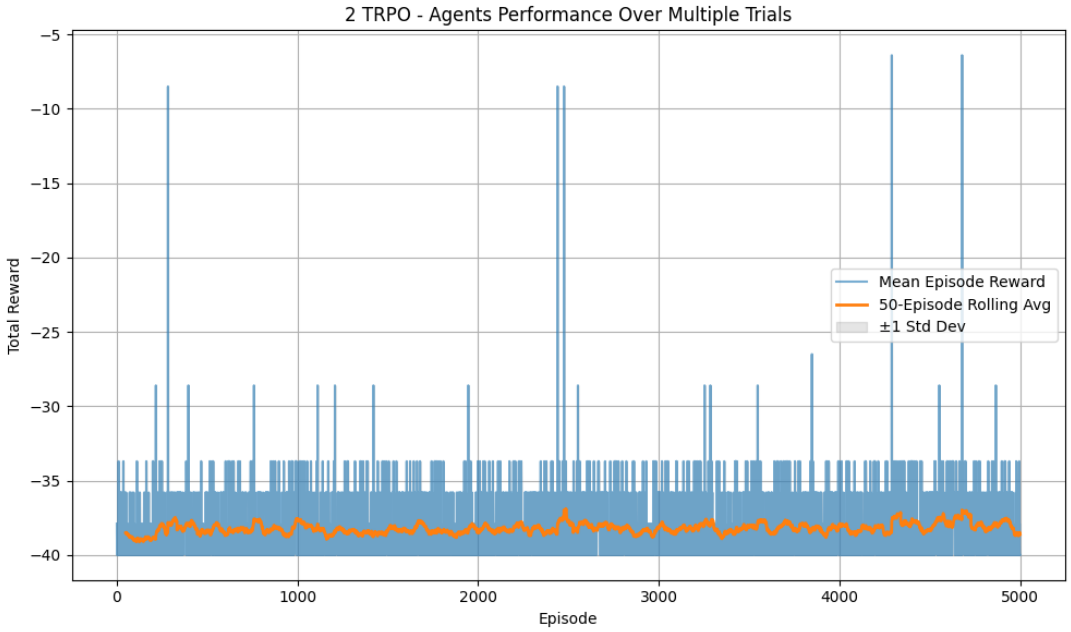
Simple layout – 2 PPO

Evaluation: gamma=0.95, lr=1e-3, eps_clip=0.2, K_epochs=8



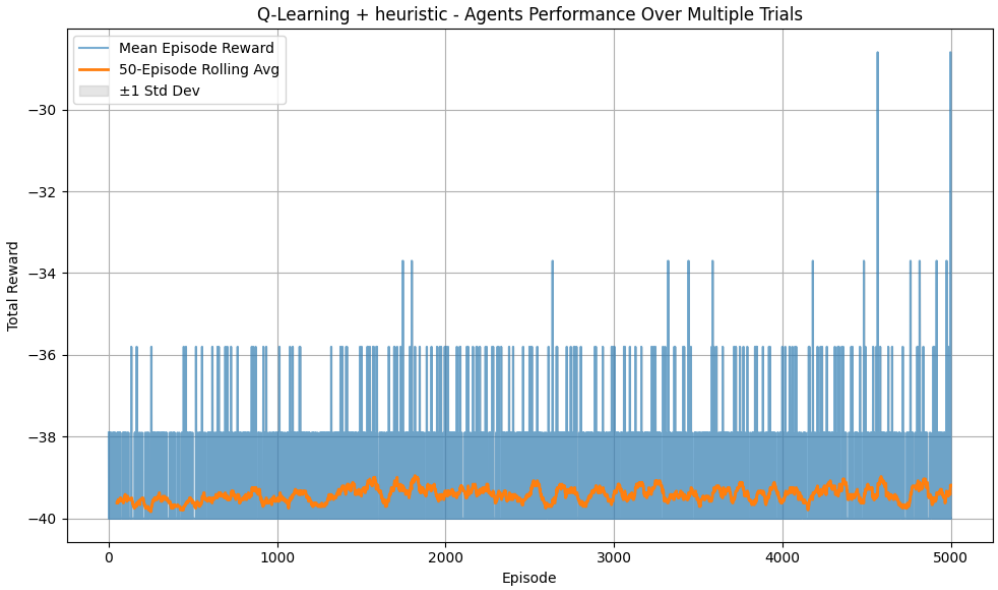
Simple layout – 2 TRPO

Evaluation: gamma = 0.99, max_kl = 1e-2, cg_iters = 10, damping = 1e-2, ls_max_steps = 10, ls_backtrack_coeff = 0.9, lr = 1e-3, update_epochs = 5



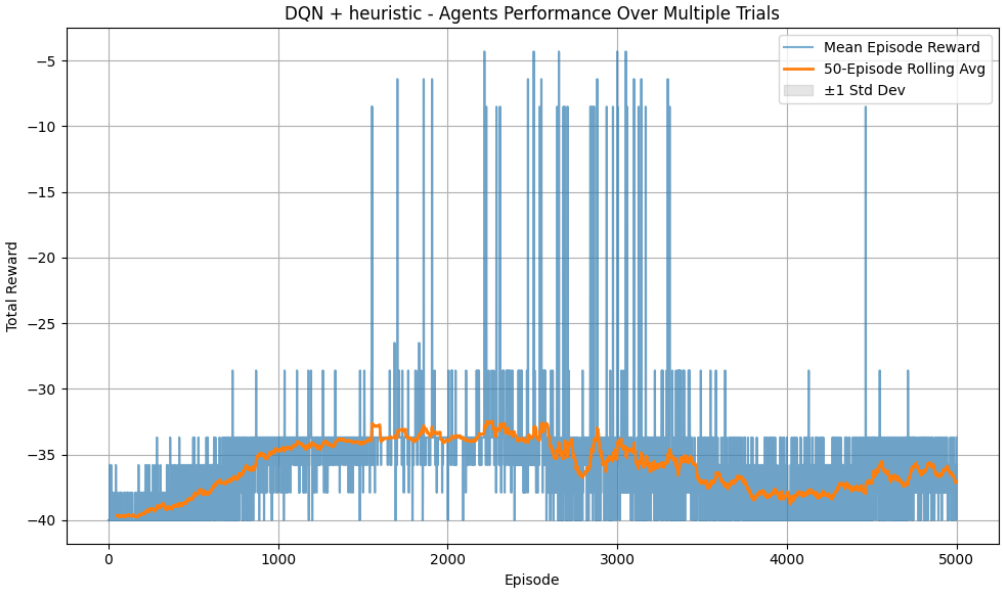
Complex layout - Q-Learning + heuristic

Evaluation: alpha=0.1, gamma=0.99, epsilon=0.1



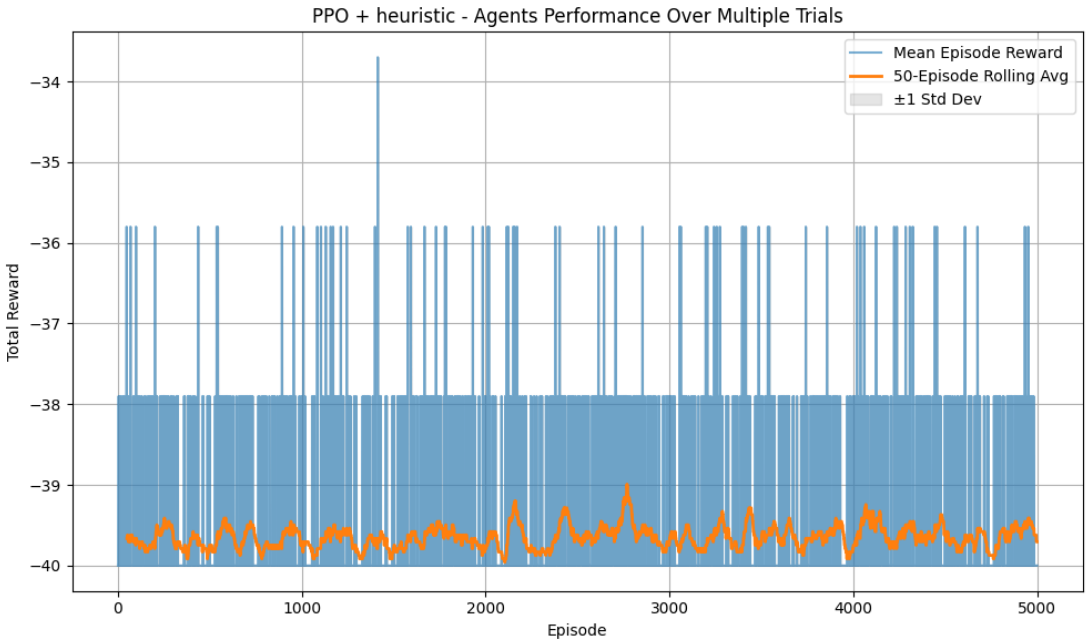
Complex layout - DQN + heuristic

Evaluation: gamma = 0.99, epsilon = 1, epsilon_decay = 0.995, min_epsilon = 0.1,batch_size = 128, update_target_every = 20, lr = 1e-4



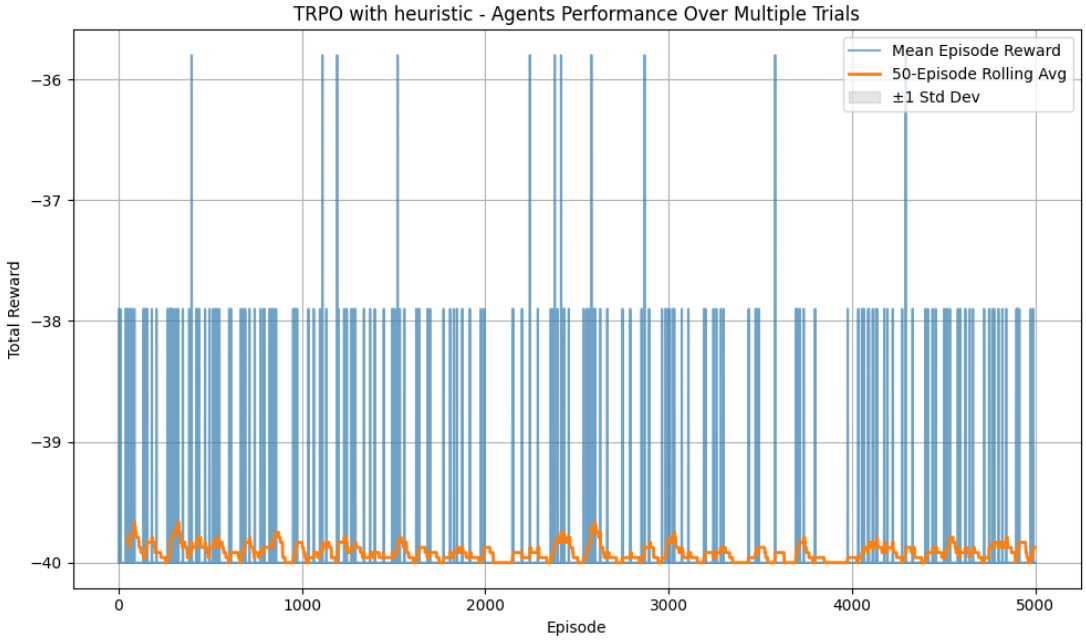
Complex layout - PPO + heuristic

Evaluation: gamma=0.95, lr=1e-4, eps_clip=0.05, K_epochs=6, entropy_coef = 0.01



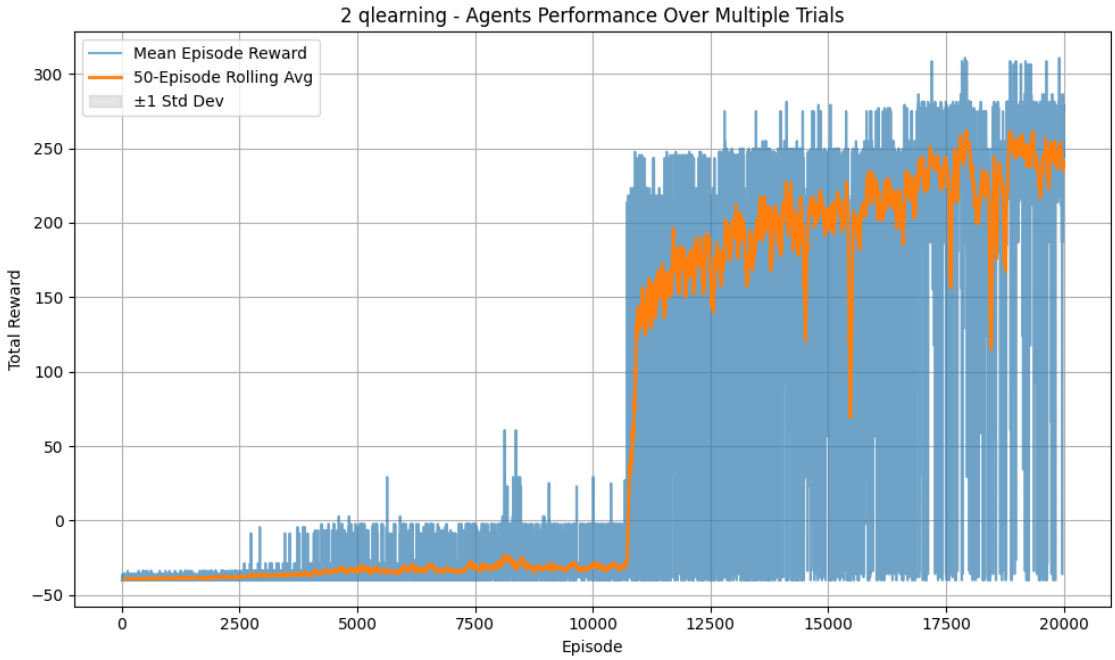
Complex layout - TRPO + heuristic

Evaluation: gamma=0.99, max_kl=1e-2, cg_iters=10, damping=1e-2, ls_max_steps=10, ls_backtrack_coeff=0.8, lr=1e-4, update_epochs=10



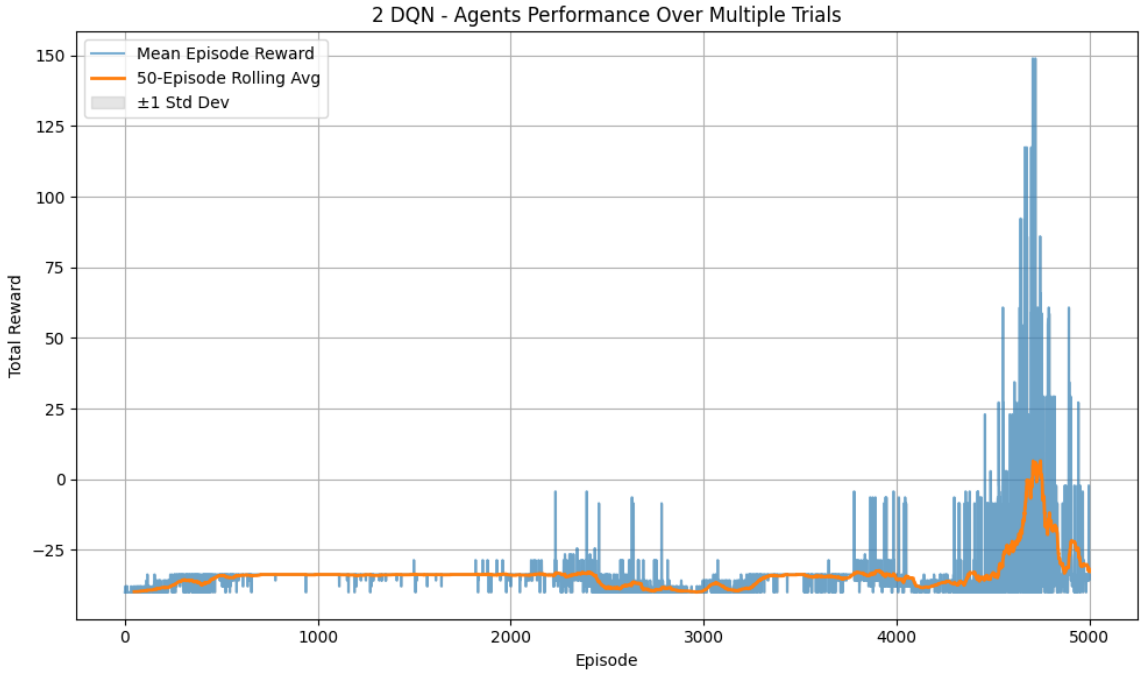
Complex layout – 2 Q learning

Evaluation: alpha=0.1, gamma=0.99, epsilon=0.1



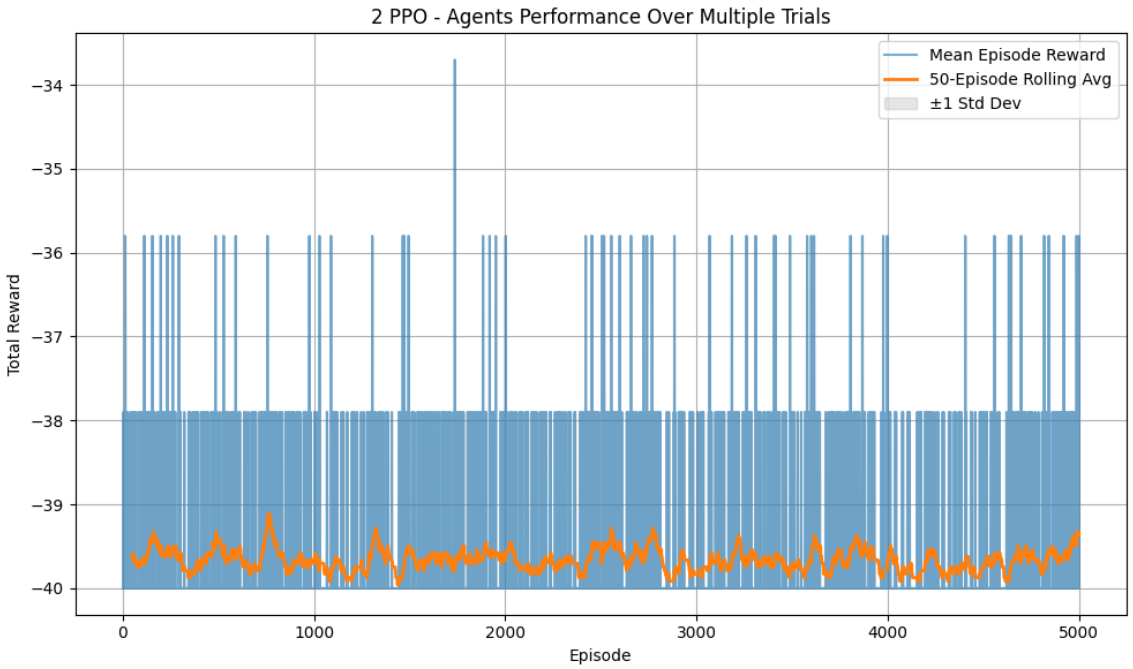
Complex layout – 2 DQN

Evaluation: gamma = 0.99, epsilon = 1.0, epsilon_decay = 0.995, min_epsilon = 0.2, batch_size =128, update_target_every = 20, lr=1e-4



Complex layout – 2 PPO

Evaluation: gamma=0.99, lr=1e-3, eps_clip=0.1, K_epochs=8



Complex layout – 2 TRPO

Evaluation: gamma = 0.99, max_kl = 1e-4, cg_iters = 10, damping = 1e-3, ls_max_steps = 10, ls_backtrack_coeff = 0.8, lr = 1e-3, update_epochs = 6