Human Factors in Design, Engineering, and Computing, Vol. 238, 2026, 2646–2656
https://doi.org/10.54941/ahfe1008489

A Controlled Comparison of MARL Training-Execution Paradigms for Cooperative Multi-UAV Fire Detection

Tanvi Chanchalani and Fuhua Lin

Athabasca University, Athabasca, AB, Canada

ABSTRACT

This paper presents a controlled empirical comparison of three fundamental multi-agent reinforcement learning (MARL) paradigms, the Centralized Training and Execution (CTE), Decentralized Training and Execution (DTE), and Centralized Training with Decentralized Execution (CTDE), in a cooperative Unmanned Aerial Vehicle (UAV) fire-detection environment under identical task conditions. In MARL, the training and execution architecture shapes what the agents can see, what they can coordinate on, and how stable the learning problem is. Our study addressed a gap where paradigm-level comparison under controlled settings remains scarce. Results across four benchmark scenarios and a 6 to 15 agents scalability test show that CTDE may be the most reliable paradigm, winning Level-1 mission completion in three of four scenarios and Level-3 path quality in all four. CTE provides a measurable advantage over CTDE in Hard-Dynamic, where execution-time global state conditioning enables faster response to mid-episode fire spawning, but its joint action space of 4^n renders it infeasible beyond thirteen agents. DTE fails systematically due to non-stationarity, with lower completion rates across all configurations and apparent safety gains that are artefacts of coordination failure rather than genuine collision avoidance. These findings confirm that paradigm choice is a first-order design decision in cooperative MARL and that CTDE may be a promising architecture for coordination-intensive multi-agent deployments at moderate to large scales.

Keywords: Multi-agent reinforcement learning, Wildfire detection

INTRODUCTION

Multi-agent Reinforcement Learning (MARL) has emerged as a powerful framework for training groups of autonomous agents to learn cooperative behaviors in complex environments. From autonomous vehicle coordination to robotic swarm control, MARL is increasingly applied to domains where multiple agents must interact, coordinate, and adapt to achieve shared or individual objectives (Canese et al., 2021; Cui et al., 2019; Huh & Mohapatra, 2024; Li et al., 2022; Liang et al., 2025; Littman, 1994; Ning & Xie, 2024; Yadav et al., 2023; Zhang et al., 2020, 2021). The ability to deploy learning-based multi-agent systems (MAS) has expanded rapidly, driven by advances in deep learning and computational resources, with landmark applications

Received September 26, 2026; Revised October 26, 2026; Accepted November 4, 2026; Available online December 1, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

such as AlphaStar in StarCraft II and OpenAI Five in Dota 2 demonstrating that MARL can achieve superhuman performance in complex strategic environments (Huh & Mohapatra, 2024; Li et al., 2025).

Despite this progress, a fundamental gap remains in literature. While many studies focus on algorithmic innovation, developing new value factorization techniques, policy gradient methods, or communication protocols, there are fewer works that systematically investigate how the choice of training and execution architecture itself shapes learning dynamics and coordination outcomes. This architectural choice centers on three paradigms, Centralized Training and Execution (CTE), Decentralized Training and Execution (DTE), and Centralized Training with Decentralized Execution (CTDE). These paradigms have distinct trade-offs in information access, scalability, coordination potential, and deployment practicality, yet empirical comparison of all three under controlled, identical environmental conditions remains limited in literature (Amato, 2024, 2025; Huh & Mohapatra, 2024; Zhang et al., 2021).

This paper addresses this gap directly. The central research question is, given the same environment, state representation, action space, and reward function, how do CTE, DTE, and CTDE compare in terms of coordination efficiency, robustness to partial observability, scalability, and execution practicality? To answer this research question, the study designed a custom cooperative navigation environment in which groups of simulated UAVs must coordinate to detect and cover target fires while avoiding collisions and managing limited energy. This UAV fire detection domain was chosen because it captures key coordination challenges relevant to real-world MAS such as that agents must share limited sensing regions, avoid interfering with one another, work under partial observability, and scale gracefully as agent numbers increase (Canese et al., 2021; Cui et al., 2019; Li et al., 2022; Yadav et al., 2023).

Literature on trade-offs between MARL paradigms shows consistent patterns. CTE offers the strongest coordination potential because action selection can use global information allowing agents to anticipate and prevent conflicts before they occur (Amato, 2024, 2025). However, the joint observation and action spaces grow exponentially with the number of agents, making training computationally expensive and scaling impractical for large MAS (Amato, 2025; Canese et al., 2021; Liang et al., 2025; Zhang et al., 2021). DTE avoids this combinatorial explosion by allowing each agent to learn independently from local observations, and scales naturally to larger teams (Canese et al., 2021; Cui et al., 2019).

However, DTE agents experience a non-stationary environment due to changing policies of co-learners, leading to convergence difficulties, miscoordination, and inefficient team behavior (Canese et al., 2021; Ning & Xie, 2024; Zhang et al., 2021, 2024). CTDE balances these extremes by using global state access and centralized training signals such as a centralized critic to guide coordination, while deploying decentralized execution policies that depend only on local observations (Amato, 2024, 2025; Huh & Mohapatra, 2024; Yuan et al., 2023). This hybrid approach

has become dominant in cooperative MARL applications, but recent work shows that CTDE also faces unresolved challenges in information use, consensus formation, exploration, and transfer across difficult coordination tasks (Kim et al., 2024; Liu et al., 2025; Takayama & Fujita, 2026; Zhang et al., 2024). This study addresses this gap by defining a consistent set of key performance indicators and applying them uniformly across all three paradigms.

The specific research questions addressed are:

RQ1: Which MARL paradigm among CTE, DTE, or CTDE achieves the highest fire detection rate and mission completion rate across a range of environment configurations?
RQ2: Does CTDE outperform both fully centralized CTE and fully decentralized DTE in cooperative UAV fire search, and if so, under what conditions?
RQ3: How does each paradigm's performance, convergence speed, and collision rate scale as number of agents increases?
RQ4: Are results consistent across different random seeds, and do paradigm rankings hold across both static and dynamic fire scenarios?

To address these, a two-dimensional discrete-grid simulation environment called FireEnv was designed and developed in Python using Pygame for visualization and interaction. FireEnv includes configurable grid sizes, variable numbers of agents, multiple target fires, and support for both static and dynamic fire spawning. The three MARL paradigms were implemented within this framework using tabular Q-learning. While deep MARL methods now dominate large-scale applications, tabular methods remain valuable for small-to-medium scale problems where policy interpretability, fast convergence, and reproducibility are priorities (Amato, 2025; Liang et al., 2025). By keeping the simulation framework fixed and only varying the learning paradigm, the study ensured that comparisons were fair and attributable to architectural differences.

The evaluation framework distinguishes three levels of analysis. Level-1 focuses on mission success including fire detection rate, mission completion rate, and sample efficiency. Level-2 examines operational efficiency and safety including convergence speed, collision frequency, energy efficiency, and scalability. Level-3 assesses path quality and policy stability including coordination pattern analysis, deadlock identification, and learned behavior visualization. This multi-level approach supports both quantitative and qualitative evaluation of the paradigm performance.

REVIEW OF RELATED LITERATURE

MARL emerged from the need to extend reinforcement learning beyond single-agent decision making into settings where multiple agents interact in a shared environment. Early work framed this shift through Markov games which generalized the Markov decision process to multi-agent settings and showed that learning dynamics change when other agents are no longer treated as part of a stationary environment (Littman, 1994). This insight

continues to shape MARL research because the core challenges of the field, including non-stationarity, combinatorial action spaces, partial observability, and coordination, arise from the presence of multiple learning agents rather than from function approximation alone (Canese et al., 2021; Huh & Mohapatra, 2024; Liang et al., 2025).

Recent published literature shows that MARL has expanded rapidly across theory and applications, but the study of the field remains fragmented. General surveys emphasize that MARL literature is more often organized either by algorithm families, such as value-based, policy-based, and actor-critic methods, or by task types, such as cooperative, competitive, and mixed settings (Canese et al., 2021; Huh & Mohapatra, 2024; Liang et al., 2025; Ning & Xie, 2024; Zhang et al., 2021). Application-oriented reviews demonstrate that MARL has successfully been applied in domains such as communication networks, future internet infrastructure, automated vehicles, video games, robotics, and multi-robot control (Yadav et al., 2023; Li et al., 2022, 2025). Though these reviews provide broad coverage, they also indicate that direct empirical comparison of paradigm-level design choices remains limited. Much of the literature evaluates specific algorithms inside a single training paradigm, most often CTDE, instead of holding the environment constant and comparing CTE, DTE, and CTDE paradigms directly within a domain (Amato, 2024, 2025; Huh & Mohapatra, 2024; Zhang et al., 2021). This gap is at the centre of this research.

RESEARCH METHODOLOGY

This research adopted a mixed methodology approach that combined a targeted literature review with primary experiments in a custom-built MARL simulator. The central principle was controlled paradigm comparison, where the three architectures, CTE, DTE, and CTDE were evaluated within a single, identical simulation environment, with only the learning paradigm varied across conditions.

The FireEnv simulator implemented a discrete $N \times N$ grid environment in Python using NumPy and Matplotlib, with a Tkinter-based graphical interface. Agents represented abstract UAVs navigating a shared grid to detect target fires. Key parameters were configurable, including grid size (default 10×10), number of agents (default 3), fire count (default 5), maximum episode steps (default 200), fire mode (random or persistent), and a dynamic fire spawning flag. In random mode, the fire positions were drawn uniformly at the start of each episode. In persistent mode, the fire positions were fixed across episodes, enabling memorization of specific layouts. When dynamic fires were enabled, each step carried a 10 percent probability of spawning an additional fire, introducing non-stationarity that tested policy robustness (Canese et al., 2021). Agents were initialised at grid corners to maximise initial spatial coverage. An episode terminated when either all fires were detected or the step limit was reached. Detection occurred when any agent occupied the same cell as an active fire.

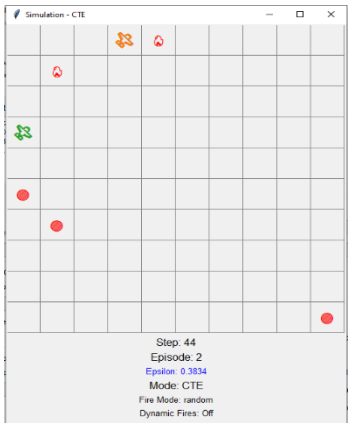


Figure 1: Screenshot of FireEnv simulation grid.

The raw grid coordinates are impractical for tabular Q-learning due to exponential state-space growth. A 3 × 3 sector decomposition is applied, mapping each agent's position to one of nine sectors. Three distinct state representations correspond to the three architectures (Amato, 2024, 2025).

For local state (DTE and CTDE), a tuple of (sector_x, sector_y, fire_info) is used, where fire_info encodes the direction and distance bucket to the nearest undetected fire. For joint state (CTE), a tuple of (agent_sectors, fire_counts) captures all agent sector assignments and per-sector fire counts. This grows in size with increasing agent count and is used to index a single shared Q-table. The aggregated joint state (CTDE) replaces per-agent sector identities with sector-level agent counts, giving a more compact coordination context. During training, each agent's Q-table is indexed by (aggregated_joint_state, local_state) while at execution time, only the local state is required.

The three architectures shared a four-action space, namely, UP, DOWN, LEFT, and RIGHT. Boundary-crossing moves left the agent stationary. Action selection used an epsilon-greedy policy throughout training (Littman, 1994). In CTE, a single controller selected a joint action over the space of 4n combinations, decoded per agent via modulo arithmetic. This exponential joint action space limited practical CTE evaluation to small team sizes and beyond ten agents it was disabled in scalability tests. In CTDE, each agent independently selected its action conditioned on the aggregated joint state via its own Q-table. In DTE, each agent selected actions using only its local state, with no information about teammates (Amato, 2025).

A six-component reward function provides a dense guidance signal while preserving the primacy of fire detection. The components are, fire detection (+15 per new fire), count-based exploration bonus ($K/\sqrt{(1+visits)}$, $K = 8.0$), row coverage (+5 for new rows visited), column coverage (+3 for new columns visited), progress toward nearest fire (+1 per unit of Manhattan distance improvement), and a collision penalty (−6 per collision). The fire detection reward is intentionally dominant, always exceeding the combined shaping bonuses in any single step. The collision penalty is deliberately soft to avoid deterring coordinated entry into high-value areas. This reward function is identical across all three paradigms (Amato, 2025; Canese et al., 2021).

The three paradigms use the standard tabular Q-learning Bellman update (Littman, 1994), $Q(s, a) \leftarrow Q(s, a) + \alpha.[r + \gamma \times \max a' Q(s', a') - Q(s, a)]$ with learning rate $\alpha = 0.3$ and discount factor $\gamma = 0.9$. These values were selected from preliminary pilot runs to balance stability and convergence speed across scenarios, rather than from exhaustive hyperparameter optimization. The architectures differ in how state, action, and reward are defined for this update. CTE uses the joint state and joint action, with reward as the mean across all agents. CTDE uses the combined (joint state, local state) key per agent with individual rewards. DTE uses only the local state per agent with individual rewards (Amato, 2024, 2025).

All three paradigms use the same reward function, episode horizon, training episodes, learning rate ($\alpha = 0.3$), discount factor ($\gamma = 0.9$), and epsilon schedule to ensure implementation fairness. Epsilon is decayed with the same horizon-aware schedule for all architectures, reaching approximately 0.10 by the final episode, with a hard floor of 0.05. Q-tables are initialized to zero in every run, so seeds do not affect initial Q-values or hyperparameter schedules. Random seeds affect stochastic environment generation only (fire layouts and dynamic fire spawning events), and the same seed is used across paradigms within each scenario. Following training, ten greedy evaluation episodes are run per mode with epsilon fixed at zero and Q-tables frozen (Amato, 2025).

Two benchmark modes are used to generate experimental results. Auto Compare evaluates all three architectures across four scenarios of increasing difficulty, using ten independent random seeds per scenario. The scenarios are, Easy-Random (8×8 grid, 3 agents, 4 fires, 80 episodes), Medium-Random (10×10, 3 agents, 5 fires, 100 episodes), Medium-Persistent (10×10, 3 agents, 5 fires, 100 episodes, fixed layout), and Hard-Dynamic (12×12, 4 agents, 7 fires, 120 episodes, dynamic spawning). The same seed produces identical stochastic environment traces for the three modes, ensuring a fair comparison. Scalability Benchmark holds the environment fixed (10×10, 5 fires) and varies agent count across {6, 7, 8}, again using ten seeds per setting, to assess whether architecture rankings at the baseline three-agent setting are preserved at larger team sizes.

A three-level framework assesses policy quality. At Level-1 is Mission Success which includes fire detection rate (fires detected / total fires), binary completion rate, and grid coverage. At Level-2 is Safety and Efficiency which includes collision count, steps to completion, and energy efficiency (fires detected per total agent steps). At Level-3 is Path Quality and Stability, which includes path gap (actual steps / oracle lower bound steps), and fire-rate standard deviation across seeds as a robustness measure. Statistical summaries (mean, median, and standard deviation) are reported across ten seeds for each metric and scenario, and level-wise winners are determined by primary metric with tie-breaking on secondary metrics.

Post-hoc statistical analysis was conducted on seed-level Auto Compare outputs (10 seeds per scenario). Two mission-success metrics were tested, completion rate and fire rate. Pairwise comparisons focused on CTDE individually versus CTE and DTE. For each comparison, Welch's t-test was used for significance testing and Cohen's d was used for effect size. To check robustness against inflated Type-I error from multiple pairwise

tests, Holm correction was applied within each metric family. This did not change the main conclusions, so the analysis reports raw Welch p-values for transparency. In the statistical analysis, smaller p-values indicate stronger evidence of a difference between methods. Positive Cohen's d indicates that CTDE performed better than the comparator, while negative values indicate the opposite.

Internal validity is supported by holding reward design, horizon, training budget, and hyperparameters constant across the paradigms. Construct validity is limited by the tabular Q-learning setting and the chosen metrics, which may not fully capture all coordination behaviors seen in deep MARL. External validity is limited by the grid-world abstraction and scenario set. Results should therefore be interpreted as paradigm-level evidence in controlled cooperative settings rather than direct field-performance guarantees.

PRIMARY RESEARCH RESULTS

The simulator results are reported as mean ± stddev over 10 independent random seeds. Each seed produces the same fire layout for the three architectures within a given scenario ensuring that observed performance differences reflect learning quality and not layout luck (Amato, 2025; Canese et al., 2021). The three-level evaluation framework applied distinguishes mission success (Level-1), operational safety and efficiency (Level-2), and path quality and policy stability (Level-3) (Amato, 2024, 2025). Tabular Q-learning is used throughout as a transparent, reproducible baseline, consistent with prior paradigm comparison research in small to medium scale cooperative environments (Liang et al., 2025; Littman, 1994).

Auto-Compare Results for Level-1 (Mission Success)

CTDE won Level-1 in three of the four scenarios, namely, Easy-Random, Med-Random, and Med-Persistent, by a substantial margin over both CTE and DTE. In Easy-Random, CTDE achieved a completion rate of 0.850 versus CTE's 0.640 while the gap widened to DTE at 0.110. This pattern is consistent with the theoretical prediction that centralized training with shared coordination context yields better cooperative policies than fully independent learning (Amato, 2024, 2025; Yuan et al., 2023). In Med-Persistent, CTDE's fire detection rate (0.906 ± 0.056) was much more stable than DTE's (0.464 ± 0.291). The high standard deviation for DTE in Med-Persistent (0.292) indicates weak policy stability under non-stationarity (Canese et al., 2021; Zhang et al., 2021). This seed-sensitivity mirrors findings in the broader MARL literature that highlight the fragility of decentralized independent learning under partial observability (Li et al., 2022, 2025; Sun et al., 2023). Hard-Dynamic was the only exception to the CTDE dominance pattern. CTE achieved a completion rate of 0.940 ± 0.084 versus CTDE's 0.920 ± 0.103, a difference of 0.020 in completion and 0.004 in fire rate (0.991 versus 0.987), with CTE also showing lower variance. This likely reflected CTE's full joint state conditioning during execution (Amato, 2025; Takayama & Fujita, 2026). This advantage, however, came at the cost of scalability as the exponential growth of CTEs joint action space (4^n) ultimately limits its applicability to small teams.

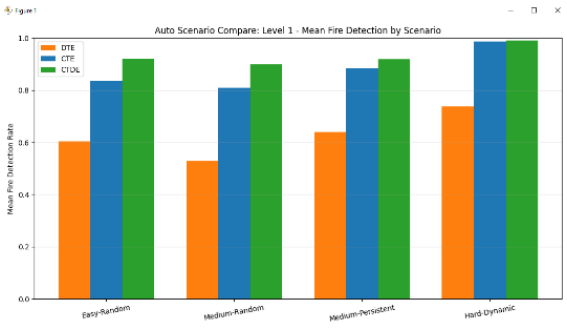


Figure 2: Auto compare results panel: Level-1 graph.

Statistical Comparison for Level-1 Metrics

Post-hoc testing of the seed-level Level-1 metrics supported the same overall interpretation as Table I. CTDE outperformed CTE in Easy-Random (completion rate: $p = .002$, $d = 1.41$; fire rate: $p = .003$, $d = 1.35$), Medium-Random (completion rate: $p < .001$, $d = 1.88$; fire rate: $p < .001$, $d = 1.93$), and Medium-Persistent (completion rate: $p = .007$, $d = 1.22$; fire rate: $p = .012$, $d = 1.12$) on both completion rate and fire rate, with statistically significant differences and large effect sizes. CTDE also outperformed DTE across all scenarios with very large effect sizes ($p < .001$, completion rate $d \geq 3.06$, fire rate $d \geq 2.17$). Hard-Dynamic was the only mixed case in which CTE was slightly higher than CTDE on both Level-1 metrics (completion rate: $p = .047$, $d = -0.89$; fire rate: $p = .016$, $d = -1.08$). The statistical evidence therefore supports CTDE as the strongest and most stable general-purpose architecture across the benchmark scenarios, while DTE remains consistently inferior as a baseline.

Auto-Compare Results for Level-2

DTE recorded the fewest collisions in the four scenarios, including zero in Hard-Dynamic. This appears to be a safety advantage, but it is an artefact of coordination failure as DTE agents, lacking any shared state representation, failed to navigate toward the same fire locations. Agents that never converge on the same grid cells can never collide. The zero-collision result is thus a symptom of the non-stationarity problem that produced DTE low completion rates (Canese et al., 2021; Zhang et al., 2021). This pattern is consistent with the MARL literature observation that independent learners often learn to avoid interaction altogether rather than learning to coordinate (Amato, 2025; Huh & Mohapatra, 2024; Li et al., 2025). On the steps taken, CTDE was the most efficient architecture with 87.5 steps in Easy-Random and 104.0 in Hard-Dynamic, versus CTE's 112.1 and 110.6 steps respectively which was approximately 7 to 28 percent longer. CTDE's efficiency advantage reflected its aggregated joint-state coordination context, which enabled agents to avoid redundant coverage without requiring full execution-time joint observation. On step efficiency, CTDE is also stronger among coordinated policies (87.5 steps in Easy-Random and 104.0 in Hard-Dynamic, versus CTE at 112.1 and 110.6), while DTE required 141.5 to 181.1 steps across scenarios. Overall, CTDE and CTE accept moderate collision counts

to achieve much higher mission completion, which is the intended objective priority for cooperative search.

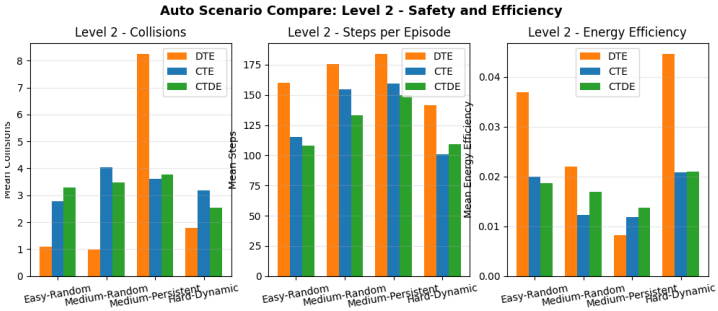


Figure 3: Level-2 graph displaying collisions and steps.

Auto-Compare Results for Level-3

CTDE achieved the smallest path gap in the four scenarios, including Hard-Dynamic where CTE won Level-1. Path gap values for CTDE range from 12.645 (Hard-Dynamic) to 20.279 (Med-Random), reflecting efficient routing relative to the oracle lower bound. CTE path gaps are in the same range in most scenarios. DTE path gaps are substantially worse as agents take more steps relative to the oracle bound because coordination failures force redundant coverage, a documented consequence of fully decentralised independent learning (Amato, 2025; Sun et al., 2023; Zhang et al., 2021). The Hard-Dynamic scenario is notable even though CTE won Level-1 and CTDE won Level-3. This indicates CTDE achieved more optimal routing overall despite its marginally lower completion rate, reflecting the route-optimisation advantage of the aggregated coordination signal during training. Policy stability (fire-rate standard deviation across 10 seeds) is lowest for CTDE throughout at 0.026 in Easy-Random, 0.056 in Med-Persistent, and 0.016 in Hard-Dynamic. DTE's instability is most in Med-Persistent (stddev 0.291) which is nearly 6 times more variable than CTDE confirming that independent learners are highly sensitive to seed-specific initialization in spatially complex, fixed-layout tasks. This instability mirrors theoretical analyses of non-stationarity in independent Q-learning, where simultaneous policy updates violate the stationarity assumptions required for Q-learning convergence guarantees.

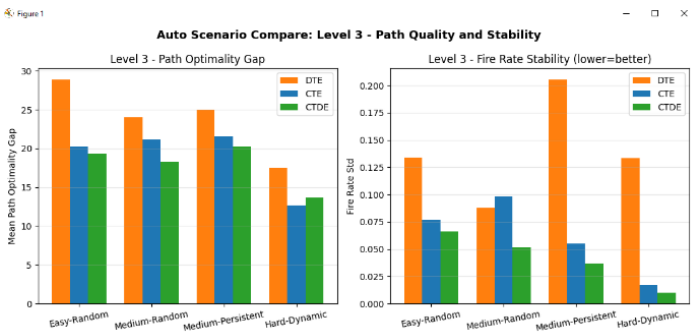


Figure 4: Level-3 graph displaying path gap and fire-rate standard deviation.

CONCLUSIONS

There are three methodological limitations to this study. First, tabular Q-learning restricts state representation to a sector-level abstraction, and the findings may not transfer directly to deep MARL architectures such as QMIX or MAPPO, where neural capacity, experience replay, and batch training introduce additional variables. Second, the CTDE implementation uses a simple aggregated joint state rather than a learned mixing network or centralized critic which in more expressive CTDE architectures could alter the relative ranking between CTDE and CTE in dynamic scenarios. Third, the FireEnv 2D discrete grid abstracts sensor noise, actuator uncertainty, and continuous flight dynamics that characterize real UAV deployments, limiting direct transferability. The conceptual limitation to the study is that the evaluation covers a single cooperative task type while paradigm performance may shift under different agent roles, competitive sub-objectives, or explicit inter-agent communication channels.

Future work could extend paradigm comparison to deep MARL implementations with learned value factorization, larger and noisier environments, multi-task transfer scenarios, and communication-augmented CTDE variants.

REFERENCES

Amato, C. (2024). "An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning," arXiv preprint arXiv:2409.03052, 2024, DOI: 10.48550/arXiv.2409.03052.

Amato, C. (2025). "An initial introduction to cooperative multi-agent reinforcement learning," arXiv preprint arXiv:2405.06161, 2025, DOI: 10.48550/arXiv.2405.06161.

Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., and Spanò, S. (2021). "Multi-agent reinforcement learning: A review of challenges and applications," Applied Sciences, vol. 11, no. 11, p. 4948, May 2021.

Chai, J., Zhao, Z., Zhu, Y., and Zhao, D. (2025). "A survey of cooperative multi-agent reinforcement learning for multi-task scenarios," in Artificial Intelligence Science and Engineering, vol. 1, no. 2, pp. 98–121, June 2025, DOI:10.23919/AISE.2025.000008.

Cui, J., Liu, Y., and Nallanathan, A. (2019). "The application of multi-agent reinforcement learning in UAV networks," in 2019 IEEE Intl. Conf. on Communications Workshops, Shanghai, China, 2019, pp. 1–6, DOI: 10.1109/ICCW.2019.8756984.

Huh, D., and Mohapatra, P. (2024). "Multi-agent reinforcement learning: A comprehensive survey," arXiv preprint arXiv:2312.10256, July 2024.

Kim, J.-B., Choi, H.-B., and Han, Y.-H. (2024). "Strangeness-driven exploration in multi-agent reinforcement learning," Neural Networks, vol. 172, p. 106149, Apr. 2024.

Li, T. et al. (2022). "Applications of multi-agent reinforcement learning in future Internet: A comprehensive survey," in IEEE Communications Surveys & Tutorials, vol. 24, no. 2, pp. 1240–1279, 2022, DOI: 10.1109/COMST.2022.3160697.

Li, Z., Ji, Q., Ling, X., and Liu, Q. (2025). "A comprehensive review of multiagent reinforcement learning in video games," in IEEE Transactions on Games, vol. 17, no. 4, pp. 873–892, Dec. 2025, DOI: 10.1109/TG.2025.3588809.

Liang, J., Miao, H., Li, K., Tan, J., Wang, X., Luo, R., and Jiang, Y. (2025). "A review of multi-agent reinforcement learning algorithms," Electronics, 14(4), p. 820, Feb. 2025.

Littman, M. L. (1994). "Markov games as a framework for multi-agent reinforcement learning," Proc. ICML.

Liu, L., Jiang, W., and Wang, Y. (2025). "Tacit learning with adaptive information selection for cooperative multi-agent reinforcement learning," in Proc. 24th Int. Conf. on Autonomous Agents and Multiagent Systems, 2025.

Ning, Z., and Xie, L. (2024). "A survey on multi-agent reinforcement learning and its application," Journal of Automation and Intelligence, vol. 3, no. 2, pp. 73–91, 2024.

Sun, L., Chang, Y.-C., Lyu, C., Shi, Y., Shi, Y., and Lin, C.-T. (2023). "Toward multi-target self-organizing pursuit in a partially observable Markov game," Information Sciences, vol. 648, p. 119475, 2023, DOI: 10.1016/j.ins.2023.119475.

Takayama, S. and Fujita, K. (2026). "CtoD-MAT: Bridging centralized and decentralized execution in multi-agent reinforcement learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, no. 48, pp. 41406–41408, 2026.

Yadav, P., Mishra, A., and Kim, S. (2023). "A comprehensive survey on multi-agent reinforcement learning for connected and automated vehicles," Sensors, 23(10), p. 4710.

Yuan, L., Zhang, Z., Li, L., Guan, C., and Yu, Y. (2023). "A survey of progress on cooperative multi-agent reinforcement learning in open environment," arXiv preprint arXiv:2312.01058, 2023, DOI: 10.48550/arXiv.2312.01058.

Zhang, J., Zhang, Y., Zhang, X. S., Zang, Y., and Cheng, J. (2024). "Intrinsic action tendency consistency for cooperative multi-agent reinforcement learning," in Proc. AAAI Conf. on Artificial Intelligence, vol. 38, no. 16, pp. 17600–17608, Mar. 2024.

Zhang, K., Yang, Z., and Başar, T. (2021). "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in Handbook of Reinforcement Learning and Control (Studies in Systems, Decision and Control, vol. 325), K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, Eds. Cham, Switzerland: Springer, 2021, pp. 321–384, DOI: 10.1007/978-3-030-60990-0_12.

Zhang, Q., Dong, H., and Pan, W. (2020). "Lyapunov-based reinforcement learning for decentralized multi-agent control," in Proc. 2nd Int. Conf. Distributed Artificial Intelligence (DAI), Nanjing, China, 2020, 55–68, DOI: 10.1007/978-3-030-64096-5_5.