iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85368

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

GridMind: Multi-Agent Navigation Using Reinforcement Learning in a 3D Environment

Manya D[1], Mugdha P[2], Nayana S[3], Pooja C[4], Akshay Nayak B S[5]
Department of Computer Science and Engineering Jawaharlal Nehru New College of Engineering (JNNCE) Shivamogga, Karnataka, India

Abstract: ***Multi-agent navigation is a fundamental problem in robotics, autonomous vehicles, warehouse automation, traffic management, and other systems in which several autonomous entities must reach individual destinations while sharing limited space. Classical Multi-Agent Path Finding (MAPF) focuses on producing collision-free paths, whereas Multi-Agent Reinforcement Learning (MARL) learns policies through interaction and can adapt to changing agent behavior. This literature paper reviews the foundations of reinforcement learning, Q-learning, Deep Q-Networks (DQN), multi-agent reinforcement learning, coordination, collision avoidance, centralized training with decentralized execution (CTDE), and learning-based multi-agent pathfinding. The literature identifies non-stationarity, partial observability, joint action complexity, reward design, communication overhead, scalability, and evaluation consistency as recurring challenges. Based on these findings, GridMind is positioned as a controlled research environment in which multiple agents learn navigation policies in a discrete grid while the same environment is rendered as an interactive 3D simula-tion. A staged approach—random baseline, Q-learning, DQN, independent multi-agent learning, and coordinated learning—provides an understandable and experimentally defensible pathway for studying learned coordination.***

Index Terms: Reinforcement Learning, Multi-Agent Reinforcement Learning, Deep Q-Network, Multi-Agent Path Finding, Collision Avoidance, Coordination, CTDE, 3D Simulation.

I. INTRODUCTION

Autonomous navigation becomes substantially more difficult when several agents operate at the same time. Each agent must move toward its own goal while accounting for obstacles and the decisions of other agents. Multi-Agent Path Finding (MAPF) formalizes a closely related problem: multiple agents must reach assigned goals without collisions, with objectives commonly involving path cost, total action cost, or makespan [1]. Reinforcement learning (RL) offers a different perspective. Instead of explicitly programming a complete path-planning policy, an agent learns action preferences from rewards obtained through interaction with an environment. Q-learning established a foundational model-free value-learning approach in which action values are improved from experience [2]. Deep reinforcement learning later extended value-based learning to larger state spaces through neural networks, most notably the Deep Q-Network (DQN) [3]. GridMind investigates whether reinforcement-learning agents can navigate a shared grid efficiently while avoiding static obstacles and other agents, and whether coordination improves performance compared with independent learning. The proposed system uses Python as the logical learning environment and Unity as the 3D visualization and demonstration layer. The objective is not to introduce a new MARL algorithm, but to provide a reproducible environment for comparing learning and coordination strategies.

II. REINFORCEMENT LEARNING FOUNDATIONS

In reinforcement learning, an agent interacts with an environment over discrete timesteps. At time t, the agent receives a state or observation s_t, chooses an action a_t, receives a reward r_t, and transitions to s_{t+1}. The objective is to learn a policy that maximizes expected cumulative reward. For navigation, the state can include the agent's location, goal location, nearby obstacles, and information about nearby agents. Actions can be discrete movements such as up, down, left, right, and stay.

A. Q-Learning

Watkins and Dayan introduced Q-learning as an incre-mental method for estimating action quality in Markovian domains [2]. The central quantity $Q(s, a)$ estimates the expected long-term value of taking action a in state s. The standard update is

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ r + \gamma \max Q(s', a') - Q(s,a)\ , \quad (1)$$

©IJRASET: All Rights are Reserved

where α is the learning rate and γ is the discount factor. Q-learning is attractive as a baseline because a small discrete grid provides an interpretable state-action representation. Its limitations become visible as the state space grows, particularly when information about multiple agents is included.

B. Exploration and Exploitation

Navigation learning requires a balance between explo-ration and exploitation. An epsilon-greedy strategy selects a random action with probability ϵ and the currently best-known action otherwise. High exploration early in training allows an agent to discover routes, while lower exploration later encourages use of learned policies. This mechanism is important in navigation because locally attractive actions can lead to dead ends or future conflicts.

C. Deep Q-Networks

Mnih et al. demonstrated that a neural network can approximate action values in a DQN, allowing reinforce-ment learning to operate on high-dimensional inputs and generalize beyond a tabular representation [3]. DQN com-monly uses an online network, a target network, experience replay, and epsilon-greedy exploration. For GridMind, the initial DQN can receive a compact numerical state vector rather than raw images. This preserves the research focus on navigation and multi-agent interaction while using the 3D environment mainly for simulation and visualization.

D. Relevance to GridMind

The literature supports a staged learning architecture. A random policy provides a lower baseline; Q-learning provides an interpretable value-based baseline; and DQN provides a neural value approximation. This progression gives measurable comparisons before multi-agent coordina-tion is introduced.

III. MULTI-AGENT REINFORCEMENT LEARNING

Multi-Agent Reinforcement Learning extends RL to environments containing multiple decision-making agents. Unlike single-agent RL, the environment is partly shaped by the behavior of other learning agents. Gronauer and Diepold survey cooperative, competitive, and mixed MARL settings and emphasize the training schemes and challenges specific to multi-agent learning [4]. Hernández-Leal et al. further discuss centralized training, decentralized execution, opponent modeling, communication, and coordination as major research directions [5].

A. Non-Stationarity

A central difficulty is non-stationarity. In single-agent RL, the environment's transition dynamics are usually treated as stable while the agent learns. In MARL, other agents continuously change their policies. From the per-spective of one learner, the behavior generating part of the environment can therefore change during training [4], [5]. This makes independent learning useful as a baseline but potentially unstable when agents strongly interact.

B. Joint Action and State Complexity

With n agents, the joint action space can grow rapidly because each timestep involves a combination of individual actions. Joint observations can also become large. The resulting complexity motivates decentralized policies, value factorization, parameter sharing, communication, and cen-tralized training [4], [5].

C. Formal Multi-Agent Problem Formulation

A useful abstraction for GridMind is a cooperative partially observable multi-agent environment. Let N agents operate in state space S. Agent i has an action set A and receives an observation o_i. A joint action at time t is

$$\mathbf{a}^t = (a_1^t, a_2^t, \ldots, a_N^t). \quad (2)$$

and the environment transitions according to $s^{t+1} \sim P(s^{t+1} \mid s^t, \mathbf{a}^t)$. Each agent follows a policy $\pi_i(a^t \mid o^t)$. For cooperative navigation, a team objective can be represented by a shared or appropriately shaped reward R^t that balances goal progress, collisions, invalid moves, and episode completion. This formulation separates the global state used for controlled experiments from the local information available to an individual decentralized policy.

D. Independent Learning

Independent learning gives each agent its own policy or value function and treats other agents as part of the environment. It is conceptually simple and provides an important baseline for GridMind. However, independent learners can struggle with the changing behavior of other agents and coordinated tasks [5]. In GridMind, independent DQN is therefore a reference point rather than the final objective.

E. Coordination

Coordination becomes necessary when the best action of one agent depends on the actions of others. Lowe et al. showed that multi-agent actor-critic methods can explicitly account for other agents and learn coordination strategies in cooperative and competitive environments [6]. Value-factorization methods such as QMIX address cooperative coordination by learning a joint action-value through a structured combination of per-agent values [7].

F. Communication

Communication is another approach to coordination. Re-cent survey work identifies communication as a mechanism through which agents can broaden their observations and improve cooperation, while also introducing design choices concerning when, with whom, and what to communicate [8]. For GridMind, communication is an optional advanced extension rather than a prerequisite for the core comparison.

IV. MULTI-AGENT PATH FINDING AND NAVIGATION

MAPF provides an important complementary research perspective. Stern et al. define MAPF as planning paths for multiple agents so that they can execute them concurrently without collisions and provide terminology and benchmark guidance for the field [1]. MAPF is especially relevant to GridMind because the proposed environment is grid-based and each agent has a start and goal location.

A. Classical MAPF Versus Learning-Based Navigation

Classical MAPF methods explicitly search for paths under constraints. They can provide strong solutions when environmental assumptions are known. Reinforcement learning instead learns a policy from repeated interaction. A learning-based policy can adapt to changes in agent behavior and environmental conditions, but training can be computationally expensive and requires careful reward design and evaluation. Recent review work on deep reinforcement learning for team-based navigation and multi-agent pathfinding emphasizes the integration of DRL with MAPF and highlights the importance of unified evaluation metrics [9]. A more recent survey covering both classical and learning-based MAPF reports substantial differences in evaluation scale and methodology across the literature, reinforcing the need for controlled and comparable experiments [10].

TABLE I
COMPARISON OF LEARNING AND NAVIGATION APPROACHES

Approach	Main idea	Strength	Limitation / GridMind role
Q-learning	Tabular value learning	Simple and interpretable	State-space growth; baseline
DQN	Neural value approximation	Handles richer state representations	Training complexity; single-agent baseline
IQL	Separate learners for agents	Simple MARL extension	Non-stationarity; MARL baseline
MADDPG	Centralized critics with decentralized policies	Models agent interaction	More complex; advanced reference
QMIX	Factorizes joint value from local values	Cooperative coordination	Structured value assumptions; CTDE reference
Comm-MARL	Learned information exchange	Can improve coordination	Communication overhead; future extension
Classical MAPF	Explicit constrained path planning	Strong collision-aware planning	Less adaptive; planning reference

B. Collision Avoidance

Collision avoidance is a core constraint in both MAPF and multi-agent navigation. In a grid environment, conflicts may occur when two agents select the same destination cell, when agents attempt to swap positions, or when an agent attempts to enter an obstacle or leave the valid grid. A useful research environment should make these events explicit and measurable.

C. Deadlock and Local Conflicts

Collision-free movement is not sufficient. Agents can also become deadlocked, for example when two agents repeatedly block each other. This creates an important distinction between safety and progress: a system may produce few collisions while still taking excessive time or failing to complete all goals. Navigation evaluation should therefore include success rate, steps, reward, collisions, and deadlock rate.

D. Navigation Metrics

Recommended evaluation metrics are success rate, aver-age cumulative reward, average steps, collision count/rate, deadlock rate, distance travelled, episode completion time, and scalability. These metrics support evaluation of both navigation quality and interaction behavior [1], [9], [10].

E. Experimental Control

Experiments should use fixed test scenarios and explicit random seeds wherever possible. Comparisons should use the same grid size, obstacle layout, starting positions, goals, and evaluation episode counts. Results should report mean performance and variability rather than a single successful demonstration. This is particularly important because recent MAPF survey work identifies differences in baseline selection, environment scale, and evaluation practice across studies [10].

V. CENTRALIZED TRAINING AND DECENTRALIZED EXECUTION

Centralized Training with Decentralized Execution (CTDE) has become one of the major paradigms in cooperative MARL. During training, agents may access information that would not be available to an individual agent during execution, such as the global state or other agents' observations. During execution, each agent acts using its local observation. Surveys describe CTDE as a practical response to non-stationarity and partial observ-ability [4], [5].

For GridMind, a centralized training process could observe the positions and goals of all agents and use that information to improve learning. After training, each agent could execute using only its own observation and local information. This structure is attractive because the simulator can expose information useful for learning without requiring a centralized controller at deployment time.

A. Value Factorization

QMIX is a representative value-based CTDE approach. It learns per-agent values from local observations and combines them through a mixing network to estimate a joint action-value, with a monotonicity constraint that preserves consistency between centralized and decentralized action selection [7].

B. Actor-Critic Approaches

MADDPG is another influential multi-agent approach. Lowe et al. addressed non-stationarity by using central-ized critics that can account for other agents' actions during training while maintaining separate policies [6]. The broader literature shows that CTDE can be implemented through value-based or actor-critic approaches depending on task and observation structure [5].

C. Implications for GridMind

CTDE is technically relevant to GridMind, but it should not be the first implementation target. A sensible progres-sion is random navigation → Q-learning → single-agent DQN → independent multi-agent DQN → coordination → optional CTDE-inspired extension. This makes it possible to identify whether improvements come from the basic RL model or from multi-agent coordination.

D. Research Challenges

The literature continues to identify scalability, stabil-ity, generalization, communication, sample efficiency, and evaluation as open issues in MARL [4], [5], [8]. Recent MAPF surveys also emphasize the need for standardized and comparable evaluation [9], [10]. These observations support controlled experiments in GridMind rather than relying only on a visual demonstration.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. 3D SIMULATION ENVIRONMENTS FOR NAVIGATION RESEARCH

The proposed GridMind system differs from a purely algorithmic MAPF benchmark because the final demon-stration is intended to look like a 3D simulation. The 3D engine renders the grid as a physical-looking arena containing robots, walls, obstacles, target areas, lighting, camera views, and interface elements.

The logical environment represents the navigation prob-lem as a discrete grid. Each cell is free or occupied, and each agent has a discrete position and goal. The Unity layer renders the same state as a 3D scene. This separation keeps the RL problem tractable while making the result visually demonstrable.

A. Why Not Begin With Raw Visual Input?

DQN can learn from high-dimensional sensory input, as demonstrated by Mnih et al. [3]. However, using raw images in GridMind would introduce a separate computer-vision problem. The project's primary research question concerns multi-agent navigation and coordination, not visual percep-tion. A compact numerical state vector containing position, goal, nearby obstacles, and nearby agents is therefore a controlled initial representation.

B. Python–Unity Architecture

The proposed architecture separates the Python RL back-end from the Unity visualization layer. Python maintains the authoritative logical state, environment rules, rewards, DQN/MARL training, collision and deadlock evaluation, and experiment management. An integration layer transfers states and actions and maintains synchronization. Unity renders the grid, obstacles, agents, goals, camera, lighting, interface, and animation. This separation supports headless training while retaining an interactive 3D demonstration.

C. Training and Demonstration Pipeline

The recommended pipeline is: initialize the environment, run episodes, store experience, train the model, and evaluate using predefined metrics. Unity rendering can be omitted during large-scale headless training and enabled for evaluation or demonstration.

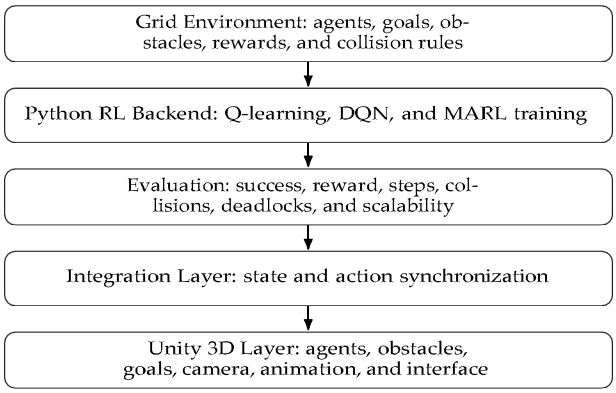


Fig. 1. Proposed GridMind architecture separating logical learning, evaluation, synchronization, and 3D visualization.

D. Research Gap

The literature contains extensive work on MARL, co-ordination, CTDE, and MAPF, but GridMind occupies a deliberately educational and integrative space: a compact grid-based navigation benchmark, explicit comparison of independent and coordinated DQN agents, controlled collision/deadlock metrics, and a 3D visualization layer. The project should not claim a new state-of-the-art MARL algorithm. Its research value is the reproducible investiga-tion of how learning and coordination affect measurable navigation outcomes.

VII. RESEARCH FRAMEWORK DERIVED FROM THE LITERATURE

Based on the reviewed literature, the proposed research framework should be developed incrementally. The envi-ronment is first validated independently from learning. A random policy establishes a baseline. Q-learning demon-strates tabular reinforcement learning. DQN replaces the table with a neural approximation. Independent DQN pro-vides the first multi-agent learning baseline. Coordination mechanisms are then introduced and evaluated against independent learning.

A. *Proposed Development Sequence*

1) Construct and test a deterministic grid-based envi-ronment with agents, goals, obstacles, rewards, and simultaneous actions.
2) Implement a random navigation baseline and estab-lish evaluation metrics.
3) Implement tabular Q-learning for a single agent.
4) Implement DQN with experience replay and a target network.
5) Extend to multiple independently learning DQN agents.
6) Implement collision detection, conflict resolution, and deadlock measurement.
7) Introduce rule-based coordination as an interpretable baseline.
8) Implement learned/coordinated navigation and com-pare it with independent DQN.
9) Optionally investigate communication or a CTDE-inspired extension after the core system is stable.
10) Render trained policies in the Unity 3D environment and conduct controlled experiments.

B. *Research Questions*

1) RQ1: Can reinforcement learning improve navigation performance over uninformed random behavior?
2) RQ2: How does DQN compare with tabular Q-learning as the navigation state becomes richer?
3) RQ3: How does independent multi-agent learning behave when agents compete for the same cells or routes?
4) RQ4: Does coordination reduce collisions and dead-locks while maintaining or improving goal completion?
5) RQ5: How does increasing the number of agents affect success, reward, collisions, and computational cost?

C. *Proposed Comparative Experiment*

The primary comparison consists of: Random as a lower baseline with no learning; Q-learning as a tabular RL baseline; DQN as a neural single-agent baseline; Inde-pendent DQN as a multi-agent baseline without explicit coordination; and Coordinated DQN as the main compar-ison for measuring the effect of coordination. CTDE or communication can be evaluated as an advanced extension after the core system is stable.

VIII. CONCLUSION

The literature establishes a progression from reinforce-ment learning to deep reinforcement learning and then to multi-agent reinforcement learning. Q-learning provides a foundational value-learning baseline [2], while DQN demon-strates neural value approximation [3]. MARL introduces additional challenges because agents learn simultaneously, creating non-stationarity, joint-action complexity, coordi-nation requirements, and communication or scalability concerns [4], [5]. MAPF provides the complementary navigation perspective in which agents must reach goals without collisions [1], while recent reviews connect deep RL with team-based navigation and emphasize consistent evaluation [9], [10]. CTDE and value-factorization methods such as QMIX provide important directions for cooperative learning [7].

For GridMind, these findings support a controlled grid-based navigation problem presented through a 3D simulation. The proposed research contribution is not a claim of a new MARL algorithm, but a structured investigation of whether learned coordination improves multi-agent navigation relative to independent learning. By combining reproducible environment rules, staged RL baselines, explicit collision and deadlock metrics, controlled experiments, and a 3D visualization layer, the project can provide a technically sound and demonstrable final-year study.

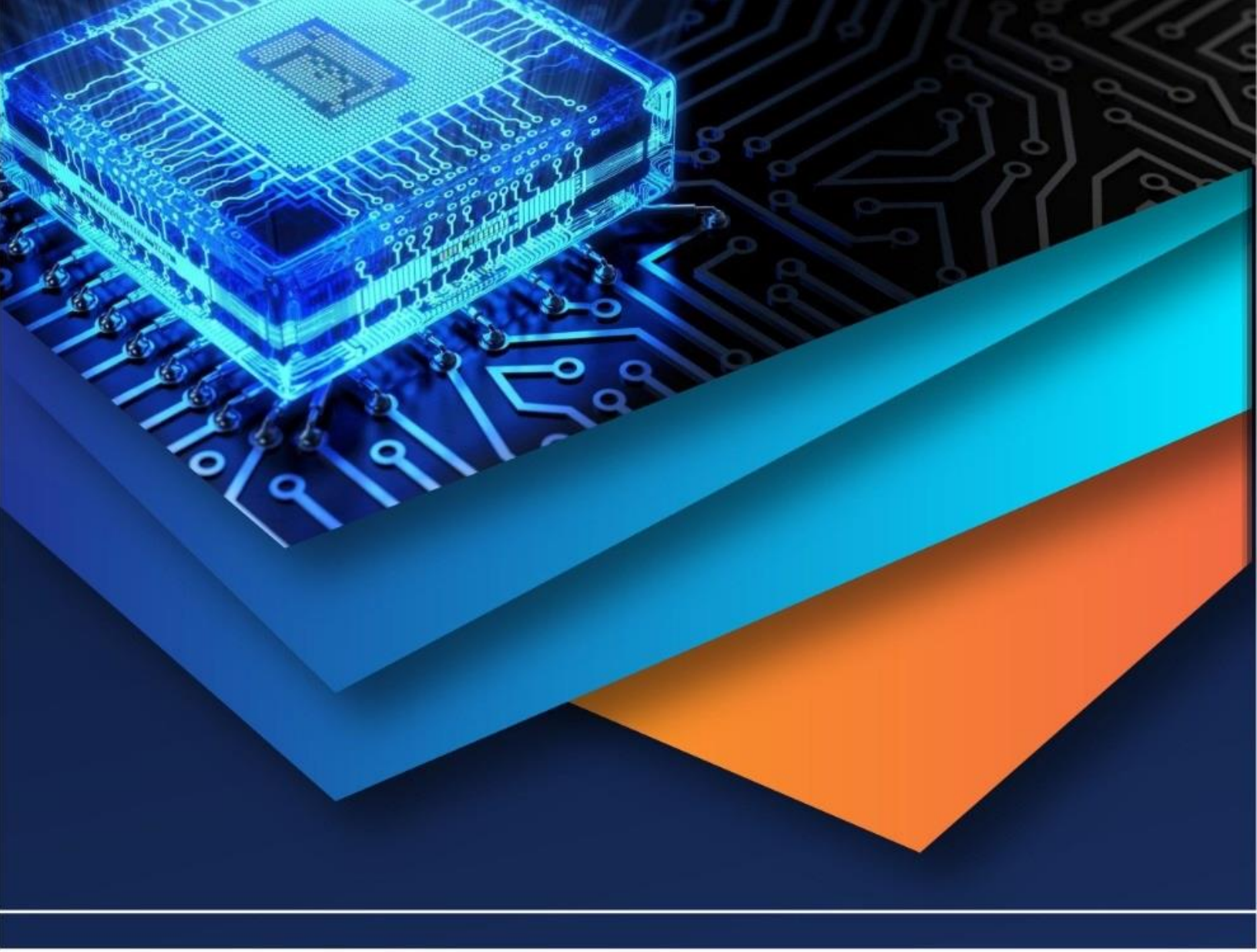
REFERENCES

[1] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. Satish Kumar, R. Bartak, and E. Boyarski, “Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks,” in *Proc. 12th Int. Symp. Combi-natorial Search (SoCS)*, vol. 10, no. 1, pp. 151–158, 2019, doi: 10.1609/SOCS.V10I1.18510.
[2] C. J. C. H. Watkins and P. Dayan, “Q-learning,” Machine Learning, vol. 8, pp. 279–292, 1992, doi: 10.1007/BF00992698.
[3] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, pp. 529–533, 2015, doi: 10.1038/na-ture14236.
[4] S. Gronauer and K. Diepold, “Multi-agent deep reinforcement learning: a survey,” Artificial Intelligence Review, vol. 55, pp. 895–943, 2022, doi: 10.1007/s10462-021-09996-w.
[5] P. Hernández-Leal, B. Kartal, and M. E. Taylor, “A survey and critique of multiagent deep reinforcement learning,” Autonomous Agents and Multi-Agent Systems, vol. 33, no. 6, pp. 750–797, 2019, doi: 10.1007/s10458-019-09421-1.
[6] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments,” in Advances in Neural Information Processing Systems, vol. 30, pp. 6379–6390, 2017.

©IJRASET: All Rights are Reserved

[7] T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning," in Proc. 35th Int. Conf. Machine Learning, PMLR, vol. 80, pp. 4295–4304, 2018.
[8] C. Zhu, M. Dastani, and S. Wang, "A survey of multi-agent deep reinforcement learning with communication," Autonomous Agents and Multi-Agent Systems, vol. 38, Art. no. 4, 2024, doi: 10.1007/s10458-023-09633-6.
[9] J. Chung, J. Fayyad, Y. Al Younes, and H. Najjaran, "Learning team-based navigation: a review of deep reinforcement learning techniques for multi-agent pathfinding," Artificial Intelligence Review, vol. 57, Art. no. 41, 2024, doi: 10.1007/s10462-023-10670-6.
[10] S. Wang, H. Xu, Y. Zhang, J. Lin, C. Lu, X. Wang, and W. Li, "Where Paths Collide: A Comprehensive Survey of Classic and Learning-Based Multi-Agent Pathfinding," arXiv:2505.19219, 2025.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)