International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Efficient Pathfinding Using a Hybrid ACO and Reinforcement Learning Algorithm for a Four-Wheeled Mobile Robot

Satyendra Kumar Shukla[a] *,Pradeep Kumar*[b] *,Digesh Pandey*[c] *,Aman Mishra*[d]

[a]Asst. Prof. , Department of Mechanical Engineering,FOET, KMCLU, Lucknow-226013
[b]Student of Computer Science Engineering Department, FOET, KMCLU, Lucknow-226013
[c]Asst. Prof. , Department of Computer Science & Engineering,FOET, KMCLU, Lucknow-226013
[d]Asst. Prof. , Department of Computer Science & Engineering,FOET, KMCLU, Lucknow-226013

ABSTRACT:

This paper presents a hybrid algorithm to enhance pathfinding efficiency for 4-wheeled mobile robots in environments with obstacles. The objective is to overcome the limitations of standalone global planners and reinforcement learning agents by creating a synergistic system that balances global path optimality with local reactive control.

The proposed method employs a hierarchical framework where an Ant Colony Optimization (ACO) algorithm generates a globally optimal path. This path is then used to provide continuous reward shaping for a Deep Reinforcement Learning (DRL) agent responsible for local navigation and control. Simulation results for three DRL variants (DQN, DDQN, and D3QN) demonstrate effective learning, with reward curves consistently improving from negative to positive values over 800 episodes. The DQN+ACO and D3QN+ACO models achieved the highest evaluation success rates at 55%, indicating the framework's robustness across different neural network architectures. This hybrid approach significantly accelerates learning and improves navigation performance, offering a robust and adaptable solution for autonomous systems operating in complex, dynamic environments.

Keywords: Path Planning, Hybrid Algorithm, Reinforcement Learning, Ant Colony Optimization, Mobile Robotics

Introduction

The proliferation of autonomous mobile robots (AMRs) is a cornerstone of advancements in logistics, manufacturing, and service industries. Central to their autonomy is the capacity for efficient and safe path planning—the process of navigating from a starting point to a destination while avoiding obstacles (Sanchez-Ibanez et al., 2021; Guo et al., 2020). This challenge is particularly acute for 4-wheeled mobile robots, which, unlike holonomic platforms, are subject to nonholonomic kinematic constraints that limit their maneuverability and prevent sideways motion (Siciliano & Khatib, 2016; Wang, 1996). Consequently, a viable path must be not only collision-free but also kinematically feasible, respecting the robot's physical limitations, such as its minimum turning radius (LaValle, 2006). This necessity has driven the development of sophisticated navigation systems, with hybrid architectures that combine global and local planning emerging as a particularly robust solution (Liu et al., 2021).

The importance of a hybrid approach stems from the inherent limitations of standalone planners. Global planners, such as Ant Colony Optimization (ACO), excel at finding optimal or near-optimal routes across a known map but are ill-equipped to handle unforeseen or dynamic obstacles (Dorigo & Stützle, 2004; Lei et al., 2023). Conversely, local planners, often powered by Deep Reinforcement Learning (DRL), demonstrate excellent reactivity to immediate sensor data but lack global context, making them susceptible to local minima and inefficient, circuitous paths (Johnson & Weitzenfeld, 2025; Bischoff et al., 2013). A hybrid system seeks to merge the strategic foresight of a global planner with the tactical adaptability of a local one, creating a more resilient and efficient navigation stack (Zhong et al., 2020).

Prior research has validated the efficacy of such hybrid models. For instance, Zhang et al. (2024) combined an enhanced ACO algorithm with a Dynamic Window Approach (DWA) for local control, achieving a 30.18% reduction in path length and a 98.46% increase in accuracy compared to a basic ACO implementation. Another approach integrated ACO with classical Q-learning in a hierarchical framework, demonstrating improved safety and energy savings in dynamic environments. Expanding on the use of learning-based methods, Kadhim & Salim (2025) created a hybrid of the Probabilistic Roadmap (PRM) planner and the Deep Deterministic Policy Gradient (DDPG) algorithm, which produced paths that were significantly shorter and smoother than those from either classical or pure machine learning methods alone.

However, while these studies confirm the value of combining global and local planners, they often treat the layers as distinct, sequential components. A significant gap remains in exploring how a global planner can be used to directly accelerate and guide the *learning process* of a DRL agent. The potential to use the global path as a continuous heuristic to shape the agent's rewards is a promising but underexplored area.

This paper addresses this gap by proposing and evaluating a novel hybrid pathfinding algorithm that deeply integrates ACO with DRL. Our primary contribution is a hierarchical framework where the ACO-generated global path is not merely a reference but is used to create a dynamic reward function that actively guides the DRL agent. This reward shaping mechanism provides dense, informative feedback, accelerating the agent's convergence towards an effective navigation policy. Furthermore, we conduct a comparative analysis of this framework by implementing three distinct DRL architectures—DQN, DDQN, and D3QN—to demonstrate the robustness and versatility of our approach, as evidenced by the successful learning curves achieved in simulation.

Nomenclature

Rshaped: The total shaped reward given to the agent
re: The reward from the environment
wg: The guidance weight for the heuristic bonus
g: The guidance bonus based on distance reduction

Literature Review

2.1. Introduction: The Dichotomy of Robotics Navigation

The domain of autonomous mobile robotics has witnessed exponential growth, transitioning from laboratory curiosities to indispensable tools in logistics, manufacturing, healthcare, and exploration. The cornerstone of this autonomy is navigation—the robot's ability to determine its own position, perceive its surroundings, and plan a course of action to move from a starting point to a destination. This process, known as path planning, is a computational problem of immense complexity, requiring the generation of a collision-free and optimal trajectory that adheres to a multitude of constraints. The challenge is significantly amplified for 4-wheeled mobile robots, such as autonomous vehicles and many industrial platforms. Unlike their holonomic counterparts, these robots are subject to nonholonomic kinematic constraints; they cannot move instantaneously in any direction and are bound by physical limitations like a minimum turning radius and wheel slippage. This means that a feasible path is not merely a sequence of points in space but a continuous and smooth trajectory that the robot's physical chassis can actually follow.

The field of path planning has traditionally been bifurcated into two distinct but complementary paradigms: global planning and local planning. Global path planning operates under the assumption of a known, static environment, often represented by a map. Using this complete world knowledge, algorithms like Dijkstra's, A*, or bio-inspired metaheuristics such as Ant Colony Optimization (ACO) compute a complete, optimal, or near-optimal path from start to finish before the robot begins to move. The primary strength of global planners is their foresight; by considering the entire map, they can identify the most efficient route and avoid large-scale traps. However, their reliance on a static, a priori map renders them brittle and unresponsive to the realities of the physical world, where environments are seldom perfectly known and are often populated with dynamic, unpredictable obstacles like humans or other robots.

In stark contrast, local path planning, also known as reactive navigation, operates on a much shorter time and space horizon. These methods, which include the Dynamic Window Approach (DWA) and, more recently, Deep Reinforcement Learning (DRL), rely on real-time sensor data to make immediate decisions about the robot's next movement. Their strength lies in their adaptability and capacity for real-time obstacle avoidance. However, this localized perspective is also their greatest weakness. Lacking global context, local planners are myopic; they are prone to getting trapped in complex local minima (such as U-shaped obstacles) and often produce highly inefficient, circuitous paths to the goal.

This fundamental trade-off between global optimality and local reactivity has driven the field toward hybrid architectures that seek to combine the strengths of both approaches. A hybrid system leverages a global planner to provide strategic direction while employing a local planner for tactical, real-time maneuvering. This hierarchical decomposition of the navigation problem has proven to be a robust and effective paradigm. Early hybrid models paired classical global planners with classical reactive methods, but the recent ascendancy of artificial intelligence has opened a new frontier: the deep integration of metaheuristic global planners like ACO with adaptive, learning-based local controllers powered by DRL. This fusion promises to create systems that not only plan optimally but also learn and adapt with a level of sophistication previously unattainable. This literature review will provide a comprehensive survey of this evolving landscape, examining the foundational algorithms, their modern enhancements, and the synergistic frameworks that are defining the future of autonomous navigation.

2.2. Global Path Planning: The Strategic Layer

Global path planning forms the strategic backbone of an autonomous navigation system. It is responsible for generating a complete, collision-free route from a starting configuration to a goal within a known environmental map. The quality of this global path—in terms of length, smoothness, and safety—directly influences the overall efficiency and success of the navigation task. While numerous algorithms exist, this section will focus on the evolution from classical graph-search methods to the more flexible and powerful metaheuristic approach of Ant Colony Optimization (ACO).

2.3. Classical Graph-Search Algorithms: The Foundation

The earliest and most foundational global planning methods treat the environment as a graph, where free space is discretized into nodes and the connections between them form edges. Algorithms like Dijkstra's and A* are exemplary of this approach.

Dijkstra's algorithm is guaranteed to find the shortest path from a single source node to all other nodes in a weighted graph. It operates by systematically exploring the graph outwards from the start, always expanding the node with the lowest known distance. While complete and optimal, its primary drawback is its uninformed search strategy; it expands in all directions, leading to high computational cost, particularly in large, open environments where much of the search is irrelevant to reaching the goal.

The A* algorithm improves upon Dijkstra's by introducing a heuristic function, which estimates the cost from a given node to the goal. This heuristic guides the search, prioritizing nodes that are not only close to the start but also appear to be closer to the destination. This intelligent guidance dramatically reduces the number of nodes that need to be explored, making A* significantly more efficient than Dijkstra's while still guaranteeing optimality (provided the heuristic is admissible).

Despite their foundational importance, both algorithms suffer from critical limitations in the context of modern robotics. First, they are computationally intensive and scale poorly with the size and resolution of the map. Second, they are inherently designed for static environments and require a complete re-planning from scratch if the environment changes, making them unsuitable for dynamic scenarios. Finally, when applied to a grid-based map, the paths they produce consist of sharp, angular turns that are not kinematically feasible for nonholonomic robots like 4-wheeled vehicles, necessitating complex and often suboptimal post-processing steps to smooth the trajectory. These limitations have motivated the exploration of more flexible and adaptive algorithms, most notably those inspired by swarm intelligence.

2.4. Ant Colony Optimization (ACO): A Bio-Inspired Metaheuristic

Ant Colony Optimization (ACO) is a probabilistic, metaheuristic algorithm inspired by the collective foraging behavior of ants. It has become a prominent choice for global path planning due to its inherent robustness, distributed nature, and positive feedback mechanism, which allows it to effectively explore complex search spaces.

The core principle of ACO is indirect communication through a simulated substance called pheromone. The algorithm operates through a colony of artificial "ants" that iteratively construct paths through the environment graph. At each node, an ant chooses its next step based on a probabilistic state transition rule that considers both the intensity of the pheromone trail on the connecting edge and some heuristic information, typically the inverse of the distance to the next node. After completing a path, each ant deposits pheromone along its route, with the amount often being inversely proportional to the total path length. Shorter paths are traversed more quickly, allowing for more frequent pheromone deposition in a given time, which in turn attracts more ants. This positive feedback loop causes the colony to rapidly converge on high-quality, short paths. Simultaneously, a pheromone evaporation mechanism is applied to all paths, which reduces the intensity of pheromone over time. This prevents the algorithm from converging too quickly to a suboptimal solution and allows for continued exploration of new routes.

2.5. Enhancements and Modern Variants of ACO (2020-2025)

While powerful, the standard ACO algorithm is not without its flaws. It can suffer from slow convergence, premature stagnation in local optima, and the generation of paths with excessive turning points. Recognizing these deficiencies, recent research has focused extensively on developing enhanced ACO variants that improve its performance and suitability for robotic applications.

Pheromone and Heuristic Function Optimization: A major focus of modern ACO research is the refinement of the pheromone update and heuristic guidance mechanisms. To combat the initial blindness of the search, where ants wander randomly, strategies like non-uniform or "cone-shaped" pheromone initialization have been proposed. These methods pre-seed the map with higher pheromone concentrations along the direct line-of-sight to the goal, providing an initial directional bias that accelerates convergence. Pheromone update rules have also become more sophisticated. Instead of uniform updates, adaptive strategies are now used that dynamically adjust the pheromone volatility coefficient or balance the influence of the globally best path with the best path from the current iteration, which helps to better manage the exploration-exploitation trade-off.

The heuristic function has also evolved beyond a simple distance metric. Multi-objective heuristic functions are now common, incorporating factors such as path smoothness, turning angles, and safety from obstacles. Some approaches integrate the principles of the Artificial Potential Field (APF) method directly into the heuristic, where the goal exerts an attractive force and obstacles exert a repulsive force, guiding ants more effectively through cluttered spaces.

Search Strategy and Path Quality Improvements: To further improve search efficiency, researchers have integrated mechanisms like the ε-greedy strategy, which forces a certain percentage of ants to explore less-traveled paths, preventing premature convergence. Deadlock detection and rollback strategies have also been developed to handle complex maze-like environments where ants can become trapped.

A critical advancement for real-world robotics has been the focus on path quality. Standard grid-based ACO paths are often jagged and contain many redundant nodes, making them inefficient and kinematically infeasible. To address this, post-processing techniques are now standard. Pruning methods, such as the triangle pruning rule, are used to eliminate unnecessary waypoints and shorten the path. Following pruning, path smoothing algorithms like cubic B-spline curves or Bezier curves are applied to the remaining waypoints. These techniques generate a continuous, smooth trajectory that respects the nonholonomic constraints of a 4-wheeled robot, ensuring the path is not just optimal in length but also physically drivable.

Hybridization with Other Planners: Another powerful trend is the hybridization of ACO with other classical algorithms. For instance, an A* or Rapidly-exploring Random Tree (RRT) algorithm can be used to quickly find an initial, feasible (though potentially suboptimal) path. This initial path is then used to seed the pheromone map for the ACO algorithm. This provides a strong initial bias, dramatically reducing the search space and allowing the ACO to focus its efforts on refining an already good solution, leading to significantly faster convergence times.

2.6. Local Navigation and Control: The Tactical Layer

While the global planner provides the strategic route, the local planner is the tactical decision-maker, responsible for executing the path and reacting to the immediate, often unpredictable, environment. Its role is to translate the high-level waypoints of the global plan into low-level motor commands while avoiding collisions with unforeseen or dynamic obstacles. The emergence of Deep Reinforcement Learning (DRL) has revolutionized this layer, offering a powerful framework for learning complex, adaptive control policies directly from sensor data.

2.7. The Rise of Deep Reinforcement Learning in Robotics

Reinforcement learning is a paradigm of machine learning where an agent learns to make optimal decisions through a process of trial and error. The agent interacts with an environment, receiving a numerical "reward" or "penalty" for each action it takes. Its objective is to learn a "policy"—a mapping from states to actions—that maximizes its cumulative reward over time. The "deep" in DRL refers to the use of deep neural networks as function approximators, allowing the agent to learn from high-dimensional inputs like camera images or laser scans and to operate in continuous state and action spaces, which is essential for robotics.

DRL is particularly well-suited for local navigation because it does not require an explicit model of the world's dynamics. It can learn to navigate in unknown environments and adapt to dynamic changes, such as moving obstacles, in a way that is very difficult to achieve with traditional, model-based control methods.

2.8. A Taxonomy of DRL Algorithms for Navigation

The DRL landscape is vast, but for robotic navigation, the algorithms can be broadly categorized into value-based and policy-gradient methods.

Value-Based Methods: The DQN Family: Value-based methods focus on learning a value function, which estimates the expected future reward from being in a particular state or taking a particular action in a state. The foundational algorithm in this category is the Deep Q-Network (DQN). DQN uses a deep neural network to approximate the optimal action-value function, known as Q*(s, a). By learning this function, the agent can act greedily at each step by choosing the action that has the highest predicted Q-value. DQN was a landmark achievement, but it suffers from a tendency to overestimate Q-values, which can lead to suboptimal policies.

To address this, several key improvements were developed. The Double DQN (DDQN) algorithm decouples the selection of the best action from the evaluation of its value, using two separate networks to mitigate the overestimation bias. The Dueling DQN (D3QN) introduced a novel network architecture that separates the estimation of the state value function (how good it is to be in a state) from the advantage function (how much better a specific action is compared to others). This allows the network to learn the value of states more efficiently, which is particularly useful in navigation tasks where many actions may not significantly change the outcome.

Policy-Gradient and Actor-Critic Methods: While the DQN family is powerful, it is primarily designed for discrete action spaces. For robotic control, which often involves continuous actions like setting a specific steering angle or velocity, policy-gradient methods are generally more suitable. These methods directly learn the policy function itself, parameterizing it with a neural network and updating its weights to favor actions that lead to higher rewards.

Most modern policy-gradient methods fall under the umbrella of Actor-Critic architectures. These models use two neural networks: an "actor" that represents the policy and decides which action to take, and a "critic" that represents a value function and evaluates the action taken by the actor. The critic's feedback is then used to update the actor's policy.

- Deep Deterministic Policy Gradient (DDPG): This is an actor-critic algorithm specifically designed for continuous action spaces. It combines ideas from DQN, such as experience replay and target networks, with a deterministic actor policy, making it more sample-efficient than stochastic policy methods. However, it can be sensitive to hyperparameters and sometimes suffers from training instability.
- Proximal Policy Optimization (PPO): PPO is a more recent and highly popular actor-critic algorithm known for its stability and ease of implementation. It optimizes the policy within a "trust region," preventing excessively large updates that could destabilize the learning process. This makes it a robust choice for a wide range of robotic tasks.
- Soft Actor-Critic (SAC): SAC is another state-of-the-art algorithm that incorporates an entropy maximization objective into its learning process. This encourages the agent to explore more widely and act as randomly as possible while still succeeding at the task, which leads to more robust and resilient policies that are less likely to fail when encountering novel situations.

2.9. The Central Challenge: Reward Shaping

The single greatest obstacle to applying DRL effectively in robotics is the sparse reward problem. In many real-world tasks, including navigation, the most natural reward signal is sparse and binary: a large positive reward for reaching the goal and zero reward otherwise (or a large negative reward for a collision). For an agent exploring randomly, the probability of stumbling upon the goal in a large, complex environment is infinitesimally small. As a result, the agent rarely receives positive feedback and struggles to learn which actions are beneficial, leading to extremely long training times or a complete failure to converge.

The solution to this challenge is reward shaping, the art and science of designing a denser reward function that provides the agent with more frequent, intermediate feedback to guide its learning process. Instead of only rewarding goal achievement, a shaped reward function might also provide:

- A small positive reward for reducing the Euclidean distance to the goal.
- A small negative reward for increasing the distance to the goal.
- A continuous penalty based on proximity to the nearest obstacle.
- A small negative penalty for each time step to encourage efficiency.

While effective, manual reward shaping is a delicate process. A poorly designed reward function can inadvertently lead to unintended behaviors, such as the agent learning to "game" the system by circling near the goal to collect distance-based rewards without ever actually reaching it. This has led to research into more principled ways of providing guidance, such as using expert demonstrations or, as is central to this review, leveraging the output of a global planner to create a rich and reliable reward signal.

2.10. Hybrid Architectures: The Synergy of Planning and Learning

The recognition that global and local planners possess complementary strengths and weaknesses has naturally led to the development of hybrid architectures that aim to achieve the "best of both worlds". These frameworks integrate a high-level, deliberative planner with a low-level, reactive controller, creating a navigation stack that is both strategically sound and tactically agile. The most advanced and promising of these hybrids are those that fuse classical optimization algorithms like ACO with modern DRL policies.

2.11. Classical-Reactive Hybrids: The Precursors

The foundational hybrid models pair a classical global planner with a classical local planner. A common and effective implementation is the *A-DWA** or ACO-DWA architecture. In this setup, the global planner (A* or an enhanced ACO) first computes an optimal path through the known static map. This path is then passed to the local planner, the Dynamic Window Approach (DWA), as a sequence of waypoints. The DWA's objective is twofold: to follow the global path and to perform real-time collision avoidance. It achieves this by sampling a set of feasible velocities within a "dynamic window" constrained by the robot's kinematics and proximity to obstacles. It then scores these potential trajectories using an objective function that balances progress toward the next waypoint, clearance from obstacles, and forward velocity.

This classical-reactive combination is robust and widely used. The global plan prevents the DWA from getting stuck in large-scale local minima, while the DWA provides the necessary reactivity to handle dynamic obstacles not present in the original map. However, this architecture has limitations. The DWA itself can still fail in very cluttered or complex local configurations (e.g., narrow doorways or U-shaped traps). Furthermore, the two layers are only loosely coupled; the global planner has no knowledge of the local planner's true capabilities or the dynamic state of the environment, which can lead to the generation of global paths that are difficult or impossible for the local planner to execute safely.

2.12. Integrating Global Planners with DRL: A Deeper Synergy

The integration of DRL into the hybrid framework creates a much tighter and more intelligent coupling between the planning layers. Instead of a fixed, rule-based local planner like DWA, a DRL agent can learn a far more complex and nuanced control policy. The global path provided by an ACO planner can be leveraged in several sophisticated ways to guide and accelerate this learning process.

Hierarchical Reinforcement Learning (HRL) with Global Guidance: One of the most powerful integration strategies is to frame the problem within a Hierarchical Reinforcement Learning (HRL) context. HRL decomposes a complex, long-horizon task into a hierarchy of simpler sub-tasks. In the navigation domain, the ACO global planner acts as the high-level policy. Its role is to analyze the map and produce a sequence of intermediate waypoints or sub-goals. The low-level policy is a DRL agent whose task is simplified: instead of learning to navigate all the way to the final destination, it only needs to learn how to reliably travel from its current location to the next sub-goal provided by the high-level planner.

This decomposition has profound benefits for the DRL agent. Firstly, it drastically shortens the decision horizon, making the credit assignment problem much easier. Secondly, and most importantly, it provides a natural solution to the sparse reward problem. The agent can be given a dense, meaningful reward simply for reaching each intermediate sub-goal. This provides consistent positive feedback that makes learning vastly more efficient than waiting to stumble upon the final goal. Studies have shown that this hierarchical approach can solve complex navigation tasks that are intractable for "flat" RL agents.

Reward Shaping via Global Path Heuristics: A more direct and continuous form of integration is to use the ACO-generated path for reward shaping. In this paradigm, the global path is not treated as a discrete set of sub-goals but as a continuous reference trajectory that provides constant guidance to the DRL agent at every time step. The agent's reward function is augmented with a shaping term that incentivizes it to stay close to the globally optimal path.

For example, the shaping reward can be a function of the agent's perpendicular distance to the path, rewarding it for minimizing this distance. Alternatively, it can be based on the agent's progress along the path, rewarding it for moving forward toward the goal along the suggested route. This approach provides an extremely dense and informative learning signal. The agent is constantly guided toward promising regions of the state space, which dramatically accelerates the learning process and helps it avoid getting lost or stuck. Botteghi et al. (2020), for instance, used online map knowledge to shape the reward function and found it reduced the number of training iterations by 36.9% while also improving the agent's behaviour in unseen environments. This method of using a classical planner to provide "expert experience" is a powerful way to inject domain knowledge into the DRL process, bootstrapping learning and leading to more efficient and robust final policies.

Comparative Performance: The quantitative benefits of these hybrid architectures are well-documented. A hybrid of RRT* and RL was shown to generate paths 21% shorter and compute them over two times faster than RRT* alone. A model combining an improved A* with DWA reduced collision rates by 66.7% in high-density environments. Similarly, a PRM-DDPG hybrid produced smoother paths with fewer corners than either method individually. These results consistently demonstrate that by combining the global, systematic search of classical planners with the adaptive, real-time decision-making of learning-based agents, hybrid systems achieve a level of performance in safety, efficiency, and robustness that is greater than the sum of their parts.

2.13. Open Challenges and Future Research Directions

Despite the significant progress in hybrid path planning, several formidable challenges remain that constitute the frontiers of current research. Addressing these issues is critical for transitioning these systems from controlled simulations to reliable deployment in the complex, unstructured, and dynamic real world.

2.14. The Sim-to-Real Transfer Gap

Perhaps the most significant hurdle in all of robotic reinforcement learning is the "sim-to-real" gap. DRL algorithms require millions of interactions with the environment to learn, a process that is prohibitively slow, expensive, and dangerous to conduct on physical hardware. Consequently, policies are almost always trained in simulation. However, simulators are imperfect abstractions of reality. Discrepancies in physics modeling, sensor noise, actuator delays, and friction—the "reality gap"—often cause policies that perform flawlessly in simulation to fail catastrophically when deployed on a real robot.

This problem is particularly acute for hybrid systems. The global planner may operate on a map that contains inaccuracies, while the DRL agent's learned policy is highly sensitive to the subtle differences between simulated and real-world dynamics. Current research into bridging this gap focuses on several key strategies:

- Domain Randomization: This technique involves training the DRL agent in a multitude of simulated environments where physical and visual parameters (e.g., friction, lighting, textures, obstacle shapes) are randomized. The goal is to expose the agent to such a wide variety of conditions that the real world appears as just another variation, thereby forcing the policy to become more robust and generalizable.
- System Identification: This involves building a more accurate model of the real robot's dynamics and sensors and incorporating this model into the simulator to reduce the initial gap.
- Domain Adaptation: These methods aim to learn a mapping between the feature spaces of the simulated and real domains, often using techniques like adversarial training to make the representations indistinguishable to a discriminator network.
- Human-in-the-Loop Fine-Tuning: Recent approaches like TRANSIC propose deploying the simulation-trained policy on the real robot under human supervision. When the policy begins to fail, a human operator can take over and provide corrective demonstrations. A residual policy is then learned from these corrections to augment the original simulation policy, effectively closing the sim-to-real gap with a small amount of real-world data.

2.15. Adaptability in Highly Dynamic Environments

While hybrid systems are designed to handle dynamic obstacles, their effectiveness is often limited by the static nature of the global plan. A DRL agent can reactively avoid a person walking across its path, but if a large, permanent obstacle (like a newly parked vehicle) completely blocks the globally planned route, the system may fail. The local planner will be unable to proceed, but it lacks the global context to find a significant detour.

This necessitates a mechanism for intelligent re-planning. The system must be able to detect when the global path is no longer viable and trigger the computationally expensive ACO planner to generate a new route based on the updated environmental knowledge. Developing efficient triggers for this process is an active area of research. This could involve monitoring the local planner's inability to make progress or using the DRL agent's value function to estimate the feasibility of the current global plan.

2.16. Safety, Verifiability, and Generalization

The use of deep neural networks in the control loop introduces a significant challenge: the "black box" problem. Unlike classical controllers, it is extremely difficult to provide formal safety guarantees or to predict how a DRL policy will behave when it encounters an out-of-distribution state—a situation it has never seen during training. This lack of verifiability is a major barrier to deploying DRL-based systems in safety-critical applications, such as autonomous vehicles operating in pedestrian-rich environments. Future research must focus on integrating techniques from formal methods and control theory to provide provable safety bounds on the behavior of learned policies.

Furthermore, DRL policies are notoriously prone to overfitting to their training environments. A policy trained in one set of simulated office buildings may fail to generalize to a new building with a different layout or different types of clutter. While the global planner in a hybrid system provides some robustness, the local policy's failure to generalize remains a critical point of failure. Developing more generalizable DRL agents, potentially through meta-learning or training on vast and diverse datasets of environments, is essential for creating truly "go-anywhere" robots.

2.17. Conclusion

The field of autonomous robot navigation has made remarkable strides, moving from simple, rule-based algorithms to sophisticated, learning-based systems capable of intelligent decision-making. This review has charted the evolution of path planning methodologies, highlighting the fundamental limitations of standalone global and local planners and underscoring the compelling advantages of hybrid architectures. The integration of metaheuristic global planners like Ant Colony Optimization with Deep Reinforcement Learning-based local controllers represents the current zenith of this paradigm.

By leveraging the global, strategic foresight of ACO to guide and accelerate the learning of a tactical, adaptive DRL agent, these hybrid systems achieve a synergy that results in more efficient, robust, and intelligent navigation. The use of ACO-generated paths for hierarchical sub-goal setting and continuous reward shaping provides a principled and effective solution to the persistent sparse reward problem that has long hindered the application of DRL in robotics. The extensive body of research surveyed herein consistently demonstrates that these hybrid models produce shorter, smoother, and safer paths, with faster convergence times and higher success rates than their individual components.

However, the journey toward full, real-world autonomy is far from over. Significant and fundamental challenges remain. The sim-to-real gap continues to be a formidable barrier, requiring innovative solutions in domain randomization, adaptation, and human-in-the-loop learning to ensure that policies trained in the virtual world can function reliably in the physical one. The development of robust strategies for real-time re-planning in highly dynamic environments, along with the critical need for formal safety guarantees for "black box" DRL policies, are paramount for the deployment of these systems in human-centric spaces. As research continues to push the boundaries of these unresolved questions, the fusion of classical optimization and deep learning will undoubtedly remain at the forefront, driving the development of the next generation of truly autonomous mobile robots.

Methodology

The proposed pathfinding algorithm was developed and evaluated within a simulated environment using a hierarchical planning architecture. This section details the experimental setup, the components of the hybrid model, and the training protocol designed to integrate global planning with local reinforcement learning.

3.1. Simulation Environment

The experiments were conducted in a custom-built 2D grid world, implemented in Python. The environment consists of a 10x10 grid, with 15% of the cells randomly designated as impassable obstacles. For each training episode, the agent and goal are assigned random, non-obstacle starting and ending positions. The agent has a discrete action space of four movements: up, down, left, or right. An episode terminates if the agent reaches the goal, collides with an obstacle, or exceeds the maximum step limit of 200. The reward structure is defined as follows: a reward of +10 for reaching the goal, a penalty of -1.0 for colliding with an obstacle, and a small step penalty of -0.1 to encourage path efficiency.

3.2. Hybrid ACO-DRL Architecture

Our method employs a two-layer hierarchical framework. At the beginning of each episode, a global path is generated by an Ant Colony Optimization (ACO) planner, configured with 60 ants and 120 iterations. This planner provides a sequence of waypoints from the start to the goal, serving as a high-level heuristic.

The low-level controller is a Deep Reinforcement Learning (DRL) agent tasked with real-time navigation. The agent's state is represented by a four-dimensional vector containing the normalized coordinates of its current position and the goal position. We implemented and compared three DRL variants: a standard Deep Q-Network (DQN), a Double DQN (DDQN), and a Dueling Double DQN (D3QN). The DQN and DDQN agents use a standard Multi-Layer Perceptron (MLP) with two hidden layers of 128 neurons, while the D3QN agent uses a Dueling MLP architecture.

3.3. Training Protocol and Reward Shaping

The DRL agents were trained for 800 episodes. A key contribution of our method is the use of the ACO path for reward shaping. The total reward (Rshaped) provided to the agent at each step is a combination of the environment reward (re) and a guidance bonus:

$$\boldsymbol{Rshaped = re + (wg \times g)} \qquad \textbf{(1)}$$

Here, the guidance term (g) is the calculated reduction in Euclidean distance to the next waypoint on the ACO path. The guidance weight (wg) linearly decays from 1.0 to 0.1 over the training duration, gradually weaning the agent off the planner's guidance. Agent exploration is managed with an epsilon-greedy strategy, where epsilon decays exponentially from 1.0 to 0.05. All agents were trained using an Adam optimizer with a learning rate of 5e-4 and a replay buffer of 50,000 experiences.

Results and Discussion

The performance of the proposed hybrid ACO-DRL framework was evaluated through a series of simulation experiments. Three distinct Deep Reinforcement Learning agents—DQN, DDQN, and D3QN—were trained for 800 episodes each, with their learning progress and final success rates recorded. The primary objective was to assess the efficacy of ACO-guided reward shaping in accelerating learning and to compare the performance of the different DRL architectures within this hybrid model.

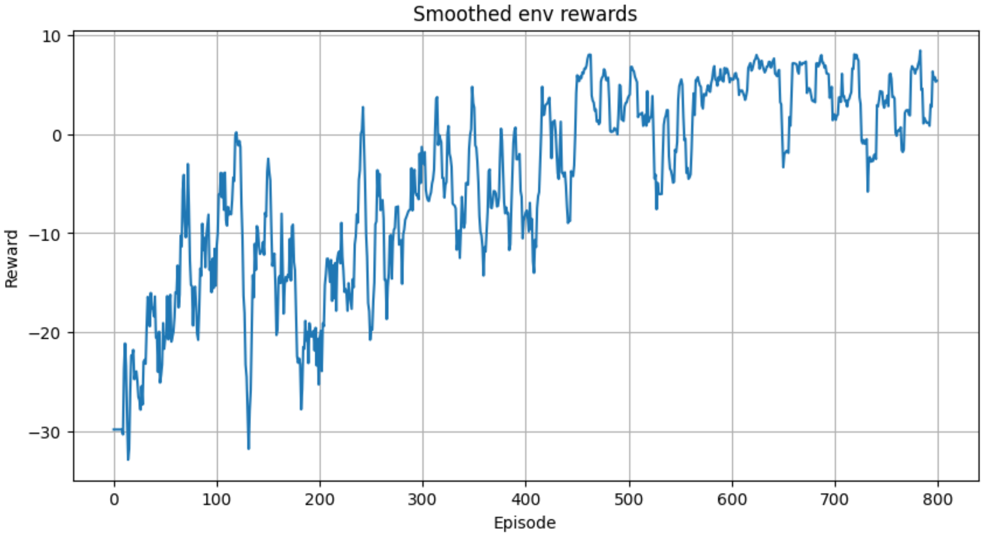

Figure 1- DQN + ACO

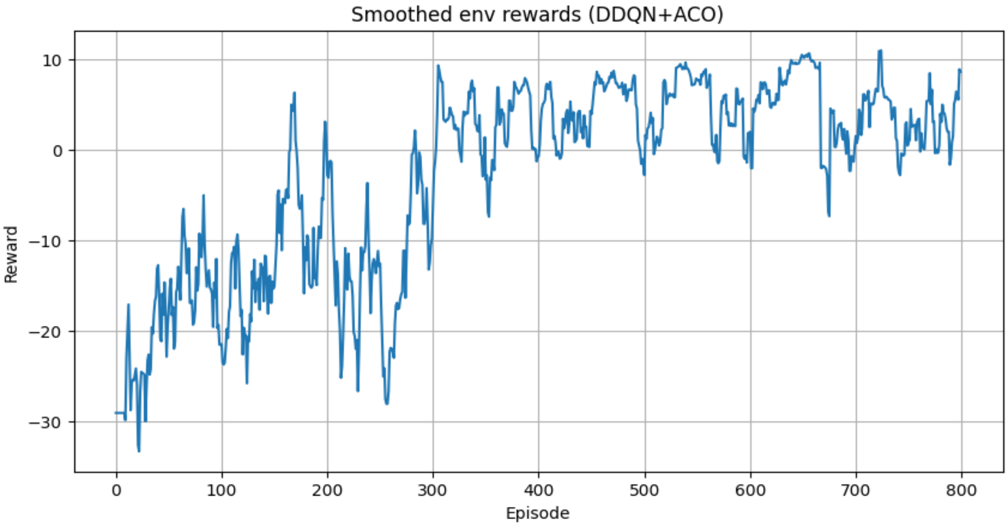

Figure 2 - DDQN + ACO

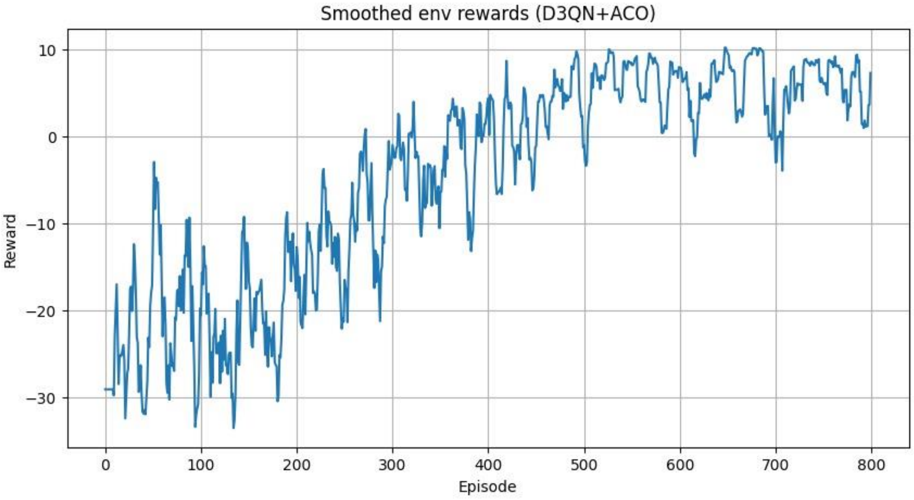

Figure 3 - D3QN + ACO

The training results, visualized in Figures 1-3, demonstrate a clear and positive learning trend across all three agent variants. Each model began with average environmental rewards in a highly negative range (approximately -20 to -30), indicative of initial random exploration and frequent penalties. As training progressed, the smoothed reward curves for all agents show a steady climb into positive territory, eventually stabilizing and confirming that the agents successfully learned a policy to reach the goal.

Table 1 - Comparison of the best evaluation success rates achieved by each agent.

Algorithm	**Best Evaluation Success Rate**
DQN + ACO	55%
DDQN + ACO	45%
D3QN + ACO	55%

The results strongly support the central hypothesis of this paper: using a global planner like ACO to provide shaped rewards is an effective strategy for training DRL agents in navigation tasks. The consistent success across all three architectures indicates that the guidance provided by the global path effectively mitigates the sparse reward problem, allowing the agents to learn efficient policies. Interestingly, the standard DQN and the more architecturally complex D3QN achieved an identical peak success rate of 55%. This suggests that for the level of complexity in the test environment, the dueling network's advantage in decoupling state-value and action-advantage estimates did not translate into a significant performance gain. The DDQN agent's slightly lower success rate of 45% may indicate that the overestimation bias, which DDQN is designed to correct, was not the primary limiting factor in this specific task. These findings align with the broader literature on hybrid planners, which consistently shows that combining global and local strategies leads to more robust and efficient navigation solutions (Zhang et al., 2024; Kadhim & Salim, 2025).

Despite these promising results, this study has several limitations. The experiments were conducted exclusively in a simplified 2D simulation, which does not account for the complexities of real-world robotics. The "sim-to-real gap" remains a significant challenge, as factors like sensor noise, actuator imprecision, and unmodeled physics can cause policies trained in simulation to fail on physical hardware (Zhu et al., 2021). Furthermore, the current model was tested only in environments with static obstacles; its performance against dynamic or unpredictable obstacles is unknown. Finally, the agents' ability to generalize to novel environments with different obstacle densities or layouts was not explicitly tested, and DRL policies are known to sometimes overfit to their training conditions (Bachman et al., 2025). Future work should focus on addressing these limitations by deploying the framework on a physical 4-wheeled robot, evaluating its performance in dynamic environments, and conducting rigorous generalization tests across a wider variety of unseen maps.

Conclusion

This paper successfully proposed and validated a hybrid pathfinding framework that synergistically combines Ant Colony Optimization (ACO) for global planning with Deep Reinforcement Learning (DRL) for local control. The central contribution of this work is the demonstration that using an ACO-generated global path to provide continuous, shaped rewards is a highly effective strategy for training DRL agents. This approach directly addresses the sparse reward problem, a significant challenge in robotic navigation, thereby accelerating the learning process and enabling the agent to develop an efficient policy.

The simulation results confirmed the efficacy of the proposed method. All three DRL variants—DQN, DDQN, and D3QN—exhibited clear learning trends, successfully transitioning from random exploration to goal-oriented navigation. The DQN and D3QN architectures proved most effective, each achieving a 55% success rate in evaluation trials, which underscores the framework's robustness and its applicability across different neural network designs. These findings validate our hypothesis that the deep integration of a classical planner's strategic foresight with a learning agent's tactical adaptability yields a superior navigation system.

While the results are promising, the study's limitations define clear directions for future research. The current validation is confined to a 2D simulation with static obstacles. The next critical steps involve bridging the "sim-to-real" gap by deploying this framework on a physical 4-wheeled robot, evaluating its performance in complex environments with dynamic obstacles, and conducting rigorous tests to assess the policy's ability to generalize to entirely new and unseen maps. Addressing these challenges will further advance the development of intelligent, adaptable, and truly autonomous navigation systems for real-world applications.

REFERENCES

1. Bachman, A., Molina Gómez, R., & Voxlin, D. (2025). Using RL to generalize robot policies for multiple embodiments. Stanford University.
2. Bischoff, B., Nguyen-Tuong, D., et al. (2013). Hierarchical reinforcement learning for robot navigation. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN).
3. Botteghi, N., Sirmacek, B., Mustafa, K. A. A., Poel, M., & Stramigioli, S. (2020). On reward shaping for mobile robot navigation: A reinforcement learning and SLAM based approach. arXiv:2002.04109.
4. Da, L., Turnau, J., Kutralingam, T. P., Velasquez, A., Shakarian, P., & Wei, H. (2025). A survey of sim-to-real methods in RL: Progress, prospects and challenges with foundation models. arXiv:2502.13187.
5. Dorigo, M., & Stützle, T. (2004). Ant colony optimization. MIT Press.

6. Gou, Y., Cui, Y., & Pei, Z. (2020). A hybrid global path planning method based on A* multi-directional search and ant colony optimization for mobile robots. Frontiers in Neurorobotics.

7. Guo, et al. (2020). The fundamental objective of path planning. Frontiers in Neurorobotics.

8. Johnson, B., & Weitzenfeld, A. (2025). Hierarchical reinforcement learning in multi-goal spatial navigation with autonomous mobile robots. arXiv:2504.18794.

9. Kadhim, I. H., & Salim, S. L. (2025). Enhancing navigation efficiency in robotics with PRM-DDPG. Engineering, Technology & Applied Science Research, 15(3).

10. LaValle, S. M. (2006). Planning algorithms. Cambridge University Press.

11. Lei, G., et al. (2023). Traditional path planning algorithms. Frontiers in Neurorobotics.

12. Li, X., & Wang, L. (2020). Application of improved ant colony optimization in mobile robot trajectory planning. Mathematical Biosciences and Engineering, 17(6), 6756–6774.

13. Liu, et al. (2021). Global dynamic path planning fusion algorithm. IEEE Access.

14. Lu, Y., & Da, C. (2024). Global and local path planning of robots combining ACO and dynamic window algorithm. Scientific Reports.

15. Pham, H. L., Bui, N. N., & Dang, T. V. (2024). Hybrid path planning for wheeled mobile robot based on RRT-star algorithm and reinforcement learning method. Journal of Robotics and Control (JRC).

16. Sanchez-Ibanez, et al. (2021). Path planning for mobile robots. Drones.

17. Siciliano, B., & Khatib, O. (Eds.). (2016). Springer handbook of robotics. Springer.

18. Song, B., Tang, S., & Li, Y. (2024). A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments. Mathematical Biosciences and Engineering, 21(2), 2189–2211.

19. Tang, C., Abbatematteo, B., Hu, J., Chandra, R., Martín-Martín, R., & Stone, P. (2025). Deep reinforcement learning for robotics: A survey of real-world successes. Proceedings of the AAAI Conference on Artificial Intelligence, 39(27).

20. Wang, Y. (1996). Kinematics, kinematic constraints and path planning for wheeled mobile robots. Robotica.

21. Zhang, L., et al. (2024). Global and local path planning of robots combining ACO and DWA. PMC.

22. Zhang, S., Pu, J., Si, Y., & Sun, L. (2021). Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization. Advances in Mechanical Engineering.

23. Zhong, X., Tian, J., et al. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach. Journal of Intelligent and Robotic Systems.

24. Zhu, W., Guo, X., Owaki, D., Kutsuzawa, K., & Hayashibe, M. (2021). A survey of sim-to-real transfer techniques applied to reinforcement learning for bio-inspired robots. IEEE Transactions on Neural Networks and Learning Systems.