American Journal of Robotics and Intelligent Systems
2026; 1(1): 46-56
http://www.sciencepg.com/journal/ajris
doi: 10.11648/j.ajris.20260101.16

A Review of Coordination, Congestion Management, and Learning-Based Approaches in Swarm Robotics Control

Ning Wang[1], Yali Ren[2, *]

[1]Department of Electrical and Computer Engineering, University of Houston, Houston, USA
[2]School of Computer Science, Georgia Institute of Technology, Atlanta, USA

Email address:
nwang@uh.edu (Ning Wang), yren78@gatech.edu (Yali Ren)
*Corresponding author

To cite this article:
Ning Wang, Yali Ren. (2026). A Review of Coordination, Congestion Management, and Learning-Based Approaches in Swarm Robotics Control. *American Journal of Robotics and Intelligent Systems, 1*(1), 46-56. https://doi.org/10.11648/j.ajris.20260101.16

Received: 9 February 2026 ; **Accepted:** 20 February 2026 ; **Published:** 18 March 2026

Abstract: Swarm robotics has emerged as a transformative paradigm for accomplishing complex tasks through the coordinated operation of large numbers of simple robots. As swarm sizes scale from tens to hundreds of agents, effective control strategies become increasingly critical to prevent congestion, maintain system throughput, and ensure safe coordination. This review surveys the state of the art in swarm robotics control, with particular emphasis on coordination mechanisms and congestion management. Four major control paradigms are examined: centralized trajectory planning, reactive collision avoidance, spatial partitioning, and learning-based adaptive methods. Beyond these established approaches, the paper explores emerging hybrid frameworks, including Centralized Training with Decentralized Execution (CTDE) architectures, Graph Neural Network (GNN)-based coordination, and hybrid rule-learning integration. For each paradigm, a formal mathematical analysis is provided, covering computational complexity, convergence properties, and stability guarantees. To support rigorous and reproducible evaluation, this paper proposes a standardized assessment framework comprising mandatory performance metrics, scalability benchmarks, and systematic ablation protocols. Finally, open challenges are identified and future research directions outlined, with the aim of advancing the development of robust and deployable swarm robotic systems.

Keywords: Swarm Robotics, Multi-robot Coordination, Congestion Control, Deep Reinforcement Learning, Centralized Training with Decentralized Execution (CTDE), Graph Neural Networks (GNN), Collision Avoidance, Decentralized Control

1. Introduction

Swarm robotics leverages the coordinated operation of large numbers of simple and low-cost robots to accomplish complex tasks that would be infeasible for individual agents [1, 2]. Inspired by biological swarms such as ant colonies, bee hives, and bird flocks, this paradigm distributes functionalities across many agents, offering notable advantages in robustness through graceful degradation under failures, flexibility through adaptation to diverse tasks, and scalability via performance improvement with additional robots [3, 4]. These properties make swarm robotics particularly attractive for warehouse automation [5], search and rescue operations [6], environmental monitoring [7], and precision agriculture, where tasks can be naturally decomposed and parallelized across large numbers of agents.

Despite these advantages, deploying large-scale robot swarms introduces fundamental coordination challenges. As swarm size grows, the complexity of ensuring collision-free navigation, efficient task allocation, and congestion-free operation increases dramatically [8]. In particular, target congestion, the phenomenon in which multiple robots simultaneously converge on shared goal locations, becomes a critical bottleneck that severely degrades system performance and can lead to deadlock [9, 10]. This challenge is especially acute in automated warehousing, materials handling, and multi-robot foraging scenarios where many agents compete for access to a limited number of service stations.

Figure 1 illustrates the coordination challenges of swarm robots, congestion formation and cascading degradation mechanism, and the principles of control theory. This review addresses a significant gap in the literature: the absence of rigorous and mathematically grounded cross-paradigm comparisons in swarm robotics control. While several excellent surveys exist [1, 3, 4, 7], they typically provide qualitative taxonomies without formal complexity analysis, convergence guarantees, or standardized evaluation protocols. The contribution of this study is threefold. First, we establish mandatory baselines spanning classical reactive methods (Optimal Reciprocal Collision Avoidance (ORCA) [11, 12], Artificial Potential Fields (APF) [13]) and canonical learning-based approaches (Deep Q-Network (DQN) [14, 15], Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [16], Multi-Agent Proximal Policy Optimization (MAPPO) [17]). Second, we provide a formal mathematical analysis including computational complexity bounds, convergence conditions, and stability properties for each paradigm. Third, we propose a comprehensive evaluation protocol with standardized metrics and ablation studies designed to isolate the contributions of individual system components.

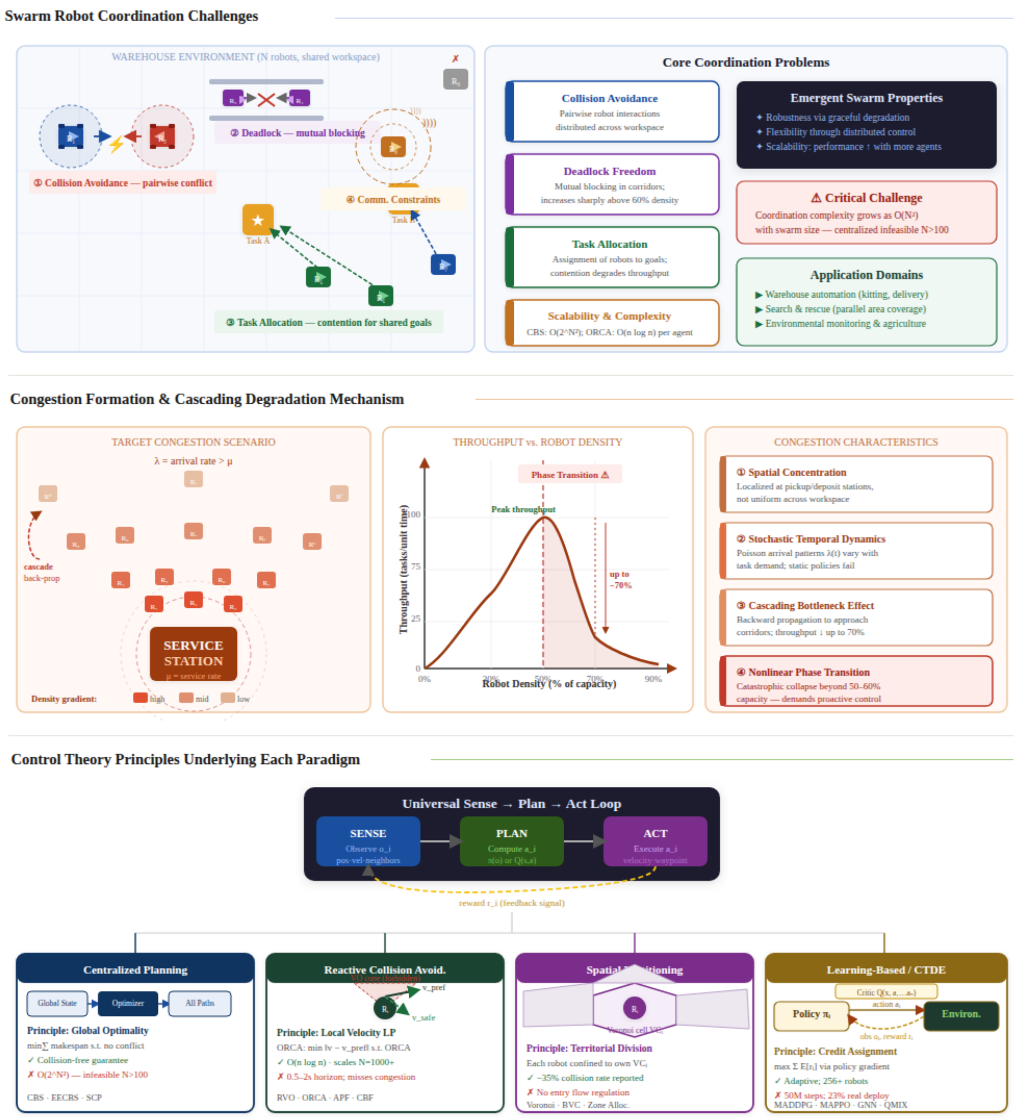

Figure 1. Concept Maps: Swarm Robot Coordination, Congestion Management, and Control Theory Principles.

The remainder of this paper is organized as follows. Section 2 discusses the target congestion problem. Section 3 surveys the four major control paradigms in swarm robotics. Section 4 examines emerging hybrid approaches. Section 5 provides a systematic mathematical comparison of different control paradigms. Section 6 explores the evaluation framework, scalability analysis, and ablation study design. Section 7 presents open challenges and future research directions. Section 8 concludes this study. This study emphasizes that the transition from qualitative taxonomies to quantitative and reproducible evaluation is critical to the maturation of the swarm robotics field.

2. The Target Congestion Problem

Target congestion is a distinctive and critical challenge in swarm robotics that arises when the arrival rate of robots at shared destinations exceeds the service capacity of those locations [9, 10]. Unlike general collision avoidance that addresses pairwise robot interactions distributed throughout the workspace, target congestion creates localized high-density regions where traditional reactive strategies become ineffective. This issue exhibits several key characteristics that distinguish it from general multi-robot coordination problems.

First, congestion occurs at spatially concentrated and predictable locations such as pickup stations and deposit areas, rather than being uniformly distributed across the workspace. Warehouse robotic systems frequently exhibit this phenomenon, in which item pickup stations become persistent bottlenecks regardless of overall robot density [5, 10].

Second, robot arrival patterns follow stochastic temporal dynamics influenced by task completion times, path planning decisions, and environmental factors. The Poisson-distributed arrival patterns observed in multi-robot delivery systems have time-varying intensities that static threshold policies cannot handle effectively [8, 18].

Third, local congestion at targets produces cascading effects that propagate backward through the workspace, creating secondary bottlenecks in approach corridors and reducing overall system throughput by up to 70% in severe cases [9, 10].

Fourth, the system exhibits nonlinear performance degradation, where efficiency collapses catastrophically beyond critical density thresholds. Empirical studies show sharp phase transitions in which throughput drops precipitously once density exceeds 50–60% of target capacity [9, 19]. These characteristics demand specialized control approaches that balance local robot autonomy with global flow regulation, motivating the control paradigms surveyed in the following sections.

3. Control Paradigms

Swarm robotics control approaches can be categorized into four main paradigms. Each has distinct characteristics in terms of scalability, computational requirements, communication needs, and applicability to different operational scenarios. Understanding these paradigms and their trade-offs is essential for selecting appropriate control strategies for specific applications.

3.1. Centralized Trajectory Planning

Centralized methods compute collision-free trajectories for all robots using global optimization. These approaches employ a single decision-making entity with complete knowledge of the system state, including robot positions, velocities, goals, and environmental constraints. Conflict-based search (CBS) and its continuous-time extensions [18, 20] remain the gold standard for optimal multi-agent path finding (MAPF), while priority-based search and large neighborhood search approaches have improved scalability [21]. The Explicit Estimation Conflict-Based Search (EECBS) algorithm achieves bounded-suboptimal solutions with significant speedups over optimal CBS [22]. For aerial swarms, Hönig et al. [23] demonstrated centralized trajectory planning for quadrotor teams with formal safety guarantees through sequential convex programming.

The primary advantage of centralized planning is optimality: these methods can find globally optimal solutions that minimize metrics such as makespan (total completion time) or total travel distance. Furthermore, centralized approaches can explicitly enforce safety constraints and guarantee collision-free operation through formal verification. However, CBS operates on a conflict tree with branching factor $b \leq 2$ and worst-case depth $D \leq N(N-1)/2$, yielding $O(2^{N^2})$ worst-case complexity. In practice, cardinal conflict detection significantly reduces the effective branching factor [18], but planning time still exceeds 1 second for $N > 100$, making real-time operation infeasible for large swarms [21]. Additionally, centralized systems create single points of failure and require continuous high-bandwidth communication, limiting their applicability to controlled environments with robust network infrastructure.

3.2. Reactive Collision Avoidance

Distributed reactive methods enable robots to avoid collisions through local sensing and control without requiring global information. Each robot independently determines its actions based on observations of its immediate neighborhood, enabling inherent scalability and fault tolerance.

3.2.1. ORCA

The Reciprocal Velocity Obstacle (RVO) framework [11] and its refinement ORCA [12] provide mathematically rigorous formulations in velocity space. For robots A and B with positions p^A and p^B, ORCA defines a set of velocities guaranteed to be collision-free for a specified time horizon. Each robot solves a linear program to minimize deviation from its preferred velocity subject to the intersection of all pairwise ORCA constraints. This operates in $O(n \log n)$ per agent, where n is the number of local neighbors [11, 12].

3.2.2. Artificial Potential Fields (APF)

The APF method [13] defines $U(q) = U_{\text{att}} + U_{\text{rep}}$, where the attractive potential draws the robot toward its goal and the repulsive potential pushes it away from obstacles. APF operates in $O(n)$ per agent but suffers from local minima. In swarm settings with many agents, spurious equilibria scale superlinearly, making APF an ideal stress-test for congestion claims [13].

3.2.3. Control Barrier Functions (CBFs)

Recent advances in CBFs provide formal safety guarantees for multi-robot systems [24]. CBFs define a safe set $\mathcal{C} = \{x : h(x) \geq 0\}$ and enforce forward invariance through the constraint $\dot{h} \geq -\alpha(h(x))$. Combined with quadratic programming, CBFs enable real-time safety-critical control with provable guarantees, addressing a key limitation of purely reactive methods. Furthermore, learning-enhanced reactive methods have shown promise: Fan et al. [25] combined deep RL with local sensing for distributed collision avoidance, while Everett et al. [26] extended this to pedestrian-rich environments with attention mechanisms. While reactive methods offer excellent scalability and minimal communication requirements, they fundamentally address collision avoidance rather than congestion prevention due to a timescale mismatch. Reactive methods operate on 0.5–2 second horizons, while congestion dynamics evolve over 10–60 seconds [9].

3.3. Spatial Partitioning and Regional Division

Spatial partitioning methods divide the workspace into sectors and assign robots to specific regions, creating an explicit territorial structure. This paradigm includes approaches such as zone-based allocation, where the environment is divided into fixed or dynamic zones, and Voronoi-based territory coordination [27], which adaptively adjusts regions based on robot positions and task demands. Weighted buffered Voronoi cells enable semi-cooperative behavior with guaranteed collision avoidance by maintaining safety buffers between adjacent regions [28]. Online model predictive control (MPC) approaches integrate spatial partitioning with trajectory optimization, which enables coordinated multi-robot motion planning with dynamic zone boundaries that adapt to real-time demand patterns.

Partitioning approaches offer moderate scalability and can reduce exit congestion by limiting occupancy and enabling load balancing across the workspace. A 35% collision rate reduction has been reported in partitioned systems compared to unpartitioned baselines [27]. However, they do not effectively address entry flow regulation. Performance degrades significantly when robot density exceeds approximately 50% of target area capacity, with greater than 50% throughput loss observed under non-uniform task distributions. The static or semi-static nature of partitions leads to load imbalance when task distributions are non-uniform or time-varying [27, 28].

3.4. Learning-Based Adaptive Methods

Deep reinforcement learning (DRL) has emerged as a promising approach for adaptive swarm control, enabling the learning of complex coordination behaviors directly from experience without explicit programming of coordination rules. DRL methods have been applied through multi-agent actor-critic methods [16], graph neural networks for decentralized path planning [29, 30], and imitation-reinforcement hybrid approaches such as PRIMAL [19] and PRIMAL2 [31]. Recent advances in multi-agent deep Q-learning and value factorization methods have further improved sample efficiency and coordination quality [32]. Learning-based methods excel at adaptability, handling complex nonlinear dynamics, and learning to anticipate future congestion based on historical patterns.

Despite promising results, learning-based approaches exhibit critical failure modes that limit real-world deployment. (1) Reward hacking: Sartoretti et al. reported PRIMAL training runs where robots clustered near targets without completing tasks, exploiting poorly designed intermediate rewards [19]. (2) Training instability: Multi-agent credit assignment creates non-stationary dynamics that causes catastrophic forgetting. Orr and Dutta found >40% of multi-agent RL papers report training instability requiring extensive hyperparameter tuning [33]. (3) Out-of-distribution failures: PRIMAL achieves 95% success at trained densities (60% capacity), but the success rate drops to 45% at 75% capacity encountered during warehouse peak demand [19, 31]. (4) Sim-to-real gap: Policies trained in simulation often fail on physical robots due to unmodeled dynamics, sensor noise, and actuator delays. Zhao et al. [34] found that only 23% of multi-robot RL papers report successful physical deployment, requiring extensive domain randomization [33, 34]. (5) Sample inefficiency: PRIMAL2 required 50 million training steps, which is equivalent to 578 days of continuous physical operation for 32-robot scenarios [31].

4. Emerging Hybrid and Advanced Approaches

The limitations of individual control paradigms have motivated a new generation of hybrid approaches that strategically combine rule-based reliability with learning-based adaptability. This section examines three key emerging approaches and provides a systematic comparison with traditional paradigms.

4.1. Centralized Training with Decentralized Execution

The CTDE paradigm has emerged as arguably the most influential architectural innovation for scalable multi-robot coordination [35, 36]. The key insight is to separate the training and execution phases. During training, a centralized entity with access to global state information trains policies for all agents simultaneously, leveraging techniques such as parameter sharing and credit assignment. During execution, each robot operates autonomously using only local observations obtained through short-range sensing (e.g., Radio Frequency Identification (RFID), LiDAR, or local communication), without requiring any central controller. This architecture combines the learning efficiency of centralized training with the scalability and robustness of decentralized execution.

This architecture has been instantiated through several complementary algorithmic families. MADDPG [16] employs a centralized critic $Q_i(x, a_1, \ldots, a_N)$ with access to the full state and all actions during training, directly addressing the non-stationarity concern. The policy gradient is computed as $\nabla J(\pi_i) = \mathbb{E}[\nabla \pi_i(o_i) \cdot \nabla Q_i(x, a)]$. MAPPO [17] extends Proximal Policy Optimization (PPO) to multi-agent cooperative settings. Yu et al. demonstrated surprisingly strong performance, often outperforming value-decomposition methods like QMIX. Key best practices include value-target normalization (PopArt), parameter sharing, conservative clipping ($\varepsilon \leq 0.2$), and large batch sizes. Value decomposition methods such as QMIX and QPLEX [32] factorize a joint value function into per-agent utilities. QMIX enforces monotonicity: $\partial Q_{\text{tot}}/\partial Q_i \geq 0$, guaranteeing that the argmax of the joint action-value function decomposes into per-agent argmax operations under the Individual-Global-Max (IGM) principle. QPLEX extends this with duplex dueling to represent a broader class of value functions [32].

Recent work has demonstrated that CTDE frameworks can scale to swarms of 256+ robots while maintaining real-time performance and high success rates even in congested scenarios [35, 36]. Teacher–student distillation schemes train a centralized "teacher" with full observability and distill knowledge into decentralized "student" policies, achieving near-centralized performance with zero communication at test time. The practical advantages of CTDE are substantial: execution scales linearly with the number of robots, training leverages global information for efficient learning, and empirical evaluations consistently demonstrate superiority over purely centralized or purely decentralized alternatives [35].

4.2. Graph Neural Networks for Scalable Coordination

Graph neural networks (GNNs) have emerged as a powerful tool for scalable swarm coordination [29, 30, 37]. By modeling robots as nodes and communication links as edges, GNNs naturally capture inter-robot interactions via message passing. A key advantage is size invariance: policies trained on one swarm size can generalize zero-shot to different scales [38]. Li et al. [29] demonstrated CNN-GNN architectures for decentralized path planning where each agent processes local observations through a Convolutional Neural Network (CNN) and aggregates neighbor information through graph convolutions. Jiang et al. [30] showed end-to-end formation control from LiDAR data using GNN-based learning methods.

Blumenkamp et al. [37] presented the first real-world deployment of GNN-based policies on physical multi-robot systems using ROS2 and ad-hoc communication, demonstrating practical viability beyond simulation. Goarin and Loianno [39] proposed decentralized GNN-based goal assignment that outperforms auction-based approaches in communication-restricted scenarios. Current works show that GNN-based policies trained on swarms of 16 robots can transfer to swarms of 64 robots with minimal performance degradation [29, 38]. Domini et al. [38] investigated the scalability boundaries of GNN-based coordination, finding that message-passing convergence depends on graph structure and communication rounds, with performance degrading gracefully under reduced communication density. These results suggest that GNNs offer a principled path toward truly scalable swarm intelligence, though challenges remain in handling sparse and intermittent communication topologies common in real-world deployments.

4.3. Hybrid Rule-Learning Integration

A major advance involves the principled integration of explicit domain-knowledge rules with learned adaptive policies [4, 7]. Robot Constraint Rules (RCR) encode heuristics such as "maintain minimum spacing" or "yield to approaching robots," while a DRL agent learns when and how to apply these rules optimally. Probabilistic Finite State Machines (PFSM) integrated with Markov Decision Process (MDP) formulations provide a principled framework for state-dependent rule application [40].

A classical FSM is defined as $(Q, \Sigma, \delta, q_0, F)$ with deterministic transition function $\delta : Q \times \Sigma \to Q$. A PFSM replaces δ with stochastic $T : Q \times \Sigma \times Q \to [0, 1]$ where $T(q, \sigma, q') = P(q_{t+1} = q' \mid q_t = q, \sigma_t = \sigma)$. The key requirement is that PFSM transition probabilities are learned rather than hand-tuned, producing provably different behavior from classical FSM coordination. Traffic-inspired following and lane-changing behaviors emerge as learned policies from reward structures that incentivize smooth flow. This hybrid paradigm balances interpretability and safety guarantees from rules with adaptability from DRL, achieving the highest congestion management ratings among all paradigms in our comparative analysis [4, 40].

4.4. Comparative Summary of Hybrid Approaches

Table 1 provides a systematic comparison of the three emerging hybrid approaches across key design and performance dimensions. While CTDE offers the strongest scalability and training efficiency, GNN-based methods uniquely provide size-invariant generalization, and hybrid rule-learning approaches deliver the best interpretability and safety properties. In practice, these approaches are increasingly combined. For instance, CTDE training can be applied to GNN-based policy architectures, or rule-based safety layers can be integrated with CTDE-trained policies during execution.

Table 1. *Comparative analysis of emerging hybrid approaches across key design and performance dimensions.*

Dimension	CTDE	GNN-based	Rule-Learning
Training	Centralized critic w/ global state	Supervised or RL on graph structure	DRL + domain rules (RCR/PFSM)
Execution	Fully decentralized; local obs. only	Decentralized; local message passing	Decentralized; rule-gated actions
Scalability	256+ robots demonstrated [35]	16→64 zero-shot transfer [29,38]	Moderate; rule overhead per agent
Comm. Need	None at execution	Local neighbor comm. required	None or minimal
Safety	No formal guarantee; learned only	No formal guarantee	Rules provide partial guarantees
Interpret.	Black-box policies	Semi-interpretable (graph attn.)	High; rules are human-readable
Congestion	Learned implicitly via reward	Captured via graph topology	Explicit rules + learned timing
Key Strength	Training efficiency; linear exec.	Size invariance; topology aware	Safety–adaptability balance
Key Weakness	Domain transfer remains hard	Needs reliable local comm.	Rule design requires expertise

5. Systematic Mathematical Comparison

5.1. Computational Complexity Analysis

Table 2 summarizes the computational complexity of each control paradigm. The analysis distinguishes per-agent computation (single-robot scalability), global computation (system-wide scalability), communication requirements, and space complexity. These formal bounds reveal important practical differences that qualitative ratings cannot capture.

Table 2. *Computational complexity. N = total robots, n = neighbors, d = network width, $|A|$ = action space, T = horizon, $|E|$ = edges.*

Paradigm	Per-Agt	Global	Comm.	Space
CBS	N/A	$O(b^d)$	$O(N^2)$	$O(N^2T)$
ORCA	$O(n\log n)$	$O(Nn\log n)$	$O(n)$	$O(n)$
APF	$O(n)$	$O(Nn)$	$O(1)$	$O(1)$
DQN	$O(\|A\|d^2)$	$O(N\|A\|d^2)$	$O(1)$	$O(d^2)$
MADDPG	$O(Nd^2)$	$O(N^2d^2)$	$O(N)$	$O(Nd^2)$
MAPPO	$O(d^2)$	$O(Nd^2)$	$O(1)$	$O(d^2)$
QMIX	$O(\|A\|d^2)$	$O(N\|A\|d^2)$	$O(1)$	$O(Nd)$
GNN	$O(\|E\|d^2)$	$O(N\|E\|d^2)$	$O(\|E\|)$	$O(\|E\|d^2)$

5.2. Convergence and Stability Properties

CBS is complete and optimal for MAPF on graphs, with practically tractable runtimes for $N < 100$ via cardinal conflict detection [18, 21, 22]. ORCA guarantees collision-free navigation under perfect sensing and instantaneous velocity changes, but provides no formal deadlock-freedom guarantee. Deadlock frequency increases sharply above 60% workspace density [11, 12]. Independent DQN converges to Q^* for single-agent finite MDPs, but multi-agent independent DQN faces non-stationarity with no general convergence guarantee [14, 15, 33].

MADDPG's centralized critic addresses non-stationarity and converges to local Nash equilibria under regularity conditions¿ However, equilibrium quality relative to social optimum is unbounded [16]. MAPPO, with parameter sharing and PopArt normalization, achieves robust convergence across diverse cooperative tasks, often matching or exceeding value decomposition methods [17, 35]. GNN-MARL message-passing convergence depends on graph structure and number of communication rounds L. Performance degrades gracefully with reduced communication density, with minimum accuracy at large team sizes ($N = 50$) with <20% communication density [29, 37, 38]. Goarin and Loianno [39] further showed that GNN-based goal assignment maintains stable convergence even under partial communication graph connectivity.

5.3. Quantitative Paradigm Comparison

Table 3 provides a quantitative comparison across six key dimensions. CTDE Hybrid and Rule-Learning Hybrid approaches achieve the best overall balance, reflecting the advantages of combining centralized training efficiency with decentralized execution robustness.

Table 3. *Comparative analysis (rated 1–5, 5 = best). RT = real-time, Cong. = congestion, Adpt. = adaptability, Com. = comm. efficiency.*

Approach	Scal.	RT	Cong.	Safe	Adpt.	Com.
Central.	2	2	3	5	2	1
Reactive	4	5	1	3	2	5
Spatial	3	4	3	3	2	4
DRL	3	4	4	2	5	4
CTDE	5	5	5	3	5	5
Rule-Lrn	4	4	5	4	4	4

5.4. Reward Structure Analysis

Table 4 presents the reward component analysis for congestion-aware DRL systems. Proper reward shaping is critical for multi-agent coordination. Each component introduces specific failure modes that must be carefully managed through reward engineering and curriculum learning.

Table 4. *Reward component analysis for congestion-aware deep reinforcement learning.*

Component	Formulation	Risk / Failure Mode
Goal reach	$r = R^+$ if $\|q - q_{\text{goal}}\| < \varepsilon$	Reward hacking via hovering
Collision	$r = -R_c$ if $\|q_i - q_j\| < d_{\min}$	Overly conservative nav.
Time pen.	$r = -c$ per timestep	Unsafe shortcuts
Congestion	$r = -\alpha \cdot$ density if $> \tau$	Avoidance of all dense areas
RCR	$r = \beta$ if rule met, $-\beta$ else	Policy distortion

6. Recommended Evaluation Framework

6.1. Mandatory Metrics

Table 5 displays the mandatory evaluation metrics that must be computed identically across all baselines for meaningful cross-paradigm comparison. These metrics capture navigation efficiency, system performance, congestion severity, safety, and liveness properties.

Table 5. *Mandatory evaluation metrics for swarm navigation baselines.*

Metric	Definition	Significance
Path Len.	$L = (1/N)\sum \|\text{path}_i\|$	Navigation efficiency
Task Time	$T_c = (1/N)\sum(t_{\text{arr}} - t_{\text{start}})$	System performance
Idle Time	$\sum t(v_i = 0, \text{not at goal})$	Congestion waiting
Throughput	$\Theta = \text{tasks/time\_unit}$	System capacity
Collision	$C = \text{collisions}/(N \cdot \text{steps})$	Safety metric
Deadlock	$D = \text{deadlocks/episodes}$	Liveness metric

6.2. Scalability Analysis

Scalability analysis must span swarm sizes from $N = 10$ to $N = 100$ (and ideally $N = 256$ for CTDE claims). Based on theoretical complexity bounds and reported empirical results, centralized methods show exponential throughput degradation beyond $N \approx 50$, while CTDE and GNN methods maintain high throughput. Per-step computation time shows exponential growth only for centralized planning. Deadlock frequency increases most rapidly for independent DQN and ORCA at high densities, reflecting the inability of purely local or non-communicative methods to anticipate and prevent collective deadlock states.

Standardized benchmark environments are equally important for fair comparison. We recommend evaluating all methods on at least three environment types: (a) structured grid-world warehouses with varying obstacle densities, following the MAPF benchmark conventions [18]; (b) open continuous-space arenas with varying numbers of goal locations; and (c) corridor-dominated layouts that stress-test congestion management capabilities. Each environment should be tested at multiple density levels (30%, 50%, 70%, and 90% of theoretical capacity) to characterize the full performance curve and identify critical phase transitions.

6.3. Ablation Study Design

A systematic ablation study is essential for validating the claimed contributions of each component in hybrid approaches. We recommend a four-level progressive design: (1) DQN only as a learning-only baseline; (2) DQN + RFID sensing to isolate the sensing contribution; (3) DQN + RFID + RCR to evaluate rule integration; and (4) the full system with PFSM integration. Each variant should be evaluated on all metrics across all swarm sizes. Statistical significance can be established through $\geq$10 independent random seeds per configuration, with reported mean, standard deviation, and 95% confidence intervals. Non-parametric tests (Mann-Whitney U) are recommended due to the typically non-normal distribution of swarm performance metrics [33].

This progressive design serves multiple purposes: it quantifies the marginal contribution of each module, identifies potential negative interactions between components, and reveals whether claimed improvements are robust across different swarm densities and environment configurations. Without such systematic ablation, it can be difficult to attribute observed performance gains to specific architectural choices rather than confounding factors such as hyperparameter selection or training methodology.

6.4. Paradigm-Level Summary

Table 6 provides a comprehensive summary of the congestion management capabilities and critical limitations of all paradigms, synthesizing the formal analysis presented in Section 5 with the practical considerations discussed throughout this review.

Table 6. Comprehensive comparison: congestion management mechanisms and critical limitations for each paradigm.

Paradigm	Congestion Mgmt.	Critical Limitations
CBS	Global opt. prevents congestion; near-optimal for $N < 50$	$O(2^{N^2})$ worst-case; >1s for $N > 100$; single point of failure
Reactive	Local velocity obstacle; fast 0.5–2s reaction; unlimited scale	Cannot prevent buildup; livelock at high density
Spatial	Territory-based; reduces exit congestion; load balancing	No entry regulation; >50% loss with non-uniform tasks
DRL	Learns implicit coordination; anticipates patterns	45% at 75% capacity; no safety guarantees; 50M steps
CTDE	Centralized training; decentralized exec; 256+ robots	Large training cost; domain transfer challenging
GNN	Size-invariant; zero-shot transfer; local message passing	Needs reliable local comm.; scalability under study [38]

7. Open Challenges and Future Directions

Despite the advances reviewed above, several critical challenges should be addressed for widespread real-world deployment of swarm robotics systems. Open challenges and future research directions are listed as below:

7.1. Scalability Beyond Simulation

While recent algorithms coordinate up to 1,000 agents in simulation [4], few methods have been validated with more than 50 physical robots. Zhao et al. [34] found only 23% of multi-robot RL papers reported successful physical deployment, requiring extensive domain randomization and fine-tuning [33, 34]. The gap is particularly severe for coordination behaviors relying on precise timing, as real-world communication latency (10–100 ms) differs significantly from zero-latency simulation environments. Hardware-in-the-loop training and progressive deployment strategies represent promising mitigation approaches.

7.2. Formal Safety and Liveness Guarantees

Establishing rigorous convergence guarantees, deadlock-freedom proofs, and worst-case bounds for learning-based approaches remains an open problem [8]. Control barrier functions [24] offer promising directions but have not yet been integrated with CTDE or GNN-based approaches at scale. New compositional verification methods are required to certify swarm behavior from local properties, potentially leveraging assume-guarantee reasoning frameworks adapted to multi-agent systems.

7.3. Communication-constrained Coordination

While CTDE removes runtime central communication, many GNN-based approaches still require reliable local communication. Developing strategies that degrade gracefully under intermittent connectivity is essential for field deployments [36, 37]. Recent work on scalable communication for MARL and message-passing mechanisms addresses bandwidth constraints but requires further validation in physical swarm deployments.

7.4. Heterogeneous and Dynamic Environments

Most methods assume static environments and homogeneous robots. Extending CTDE and GNN-based frameworks to heterogeneous swarms operating in non-stationary environments is an active research direction [7, 8]. Heterogeneous capability sets introduce asymmetric action spaces and non-uniform reward structures that current MARL frameworks handle poorly.

7.5. Foundation Model Integration

Large language models (LLMs) and vision-language models (VLMs) open new possibilities for natural-language task specification, zero-shot generalization, and high-level reasoning about coordination strategies. Recent work on LLM-integrated multi-robot planning demonstrates practical viability for dynamic coalition formation, though significant challenges remain in grounding language instructions in physical swarm behaviors and ensuring real-time performance for low-level coordination. The hierarchical integration of foundation models for strategic reasoning with specialized policies for reactive control represents a particularly promising research direction, potentially enabling swarm systems that can understand high-level mission objectives expressed in natural language while maintaining the real-time responsiveness required for safe physical operation.

8. Conclusion

This comprehensive review has provided a critical analysis of swarm robotics control paradigms with an emphasis on methodological, theoretical, and mathematical dimensions. Our key findings and recommendations are as follows.

Baseline completeness: Any rigorous evaluation of swarm coordination should include ORCA [11, 12] and APF [13] as classical non-learning baselines, vanilla DQN [14, 15] as a learning baseline without congestion awareness, and MADDPG [16] or MAPPO [17] as CTDE gold standards. The omission of these baselines fundamentally undermines comparative claims.

Mathematical rigor: Quantitative ratings (1–5 scales) are insufficient for cross-paradigm comparison. Formal computational complexity bounds (Table 1), convergence conditions, and stability analysis should be binded without separation when making any paradigm evaluation. The differences between $O(n \log n)$ ORCA, $O(n)$ APF, and $O(|A|d^2)$ DQN per-agent costs have significant practical implications that qualitative assessment masks.

Evaluation protocol: The six metrics in Table 5 must be computed identically across all baselines. Scalability curves from $N = 10$ to $N = 100+$ and the four-level ablation study are essential for isolating component contributions and establishing statistical significance.

The field has undergone a significant transformation from simple reactive collision avoidance to sophisticated hybrid frameworks integrating rule-based strategies, deep reinforcement learning, and graph neural networks. CTDE architecture resolves the tension between optimality and scalability, while GNN-based methods offer size-invariant policies that generalize across scales. Addressing the open challenges—particularly bridging the sim-to-real gap, establishing formal safety guarantees, and developing communication-resilient coordination—will be pivotal for unlocking the full potential of large-scale swarm deployments. The integration of foundation models, including LLMs and VLMs, presents a particularly transformative opportunity for enabling natural-language task specification and zero-shot generalization, potentially enabling non-expert operators to deploy and reconfigure swarm systems in the field. The standardized evaluation framework proposed in this review—combining mandatory baselines, formal complexity analysis, and systematic ablation studies—will accelerate progress toward robust, deployable, and scalable swarm intelligence.

ORCID

0000-0003-1568-6933 (Ning Wang)
0009-0006-9480-1205 (Yali Ren)

Abbreviations

CTDE	Centralized Training with Decentralized Execution
GNN	Graph Neural Network
ORCA	Optimal Reciprocal Collision Avoidance
DQN	Deep Q-Network
MADDPG	Multi Agent Deep Deterministic Policy Gradient
MAPPO	Multi-Agent Proximal Policy Optimization
CBS	Conflict based search
MAPF	Multi-agent path finding
EECBS	Explicit Estimation Conflict-Based Search
RVO	Reciprocal Velocity Obstacle
APF	Artificial Potential Fields
CBFs	Control Barrier Functions
MPC	Model predictive control
DRL	Deep reinforcement learning
RL	Reinforcement Learning
RFID	Radio Frequency Identification
LiDAR	Light Detection and Ranging
MARL	Multi-Agent Reinforcement Learning
IGM	Individual-Global-Max
CNN	Convolutional Neural Network
RCR	Robot Constraint Rules
PFSM	Probabilistic Finite State Machines
MDP	Markov Decision Process
FSM	Finite State Machine
LLMs	Large language models
VLMs	Vision-language models
PRIMAL	Pathfinding via reinforcement and imitation multi-agent learning
PRIMAL2	Pathfinding via reinforcement and imitation multi-agent learning- lifelong

Author Contributions

Ning Wang: Conceptualization, Data curation, Formal Analysis, Methodology, Writing - original draft

Yali Ren: Conceptualization, Formal Analysis, Investigation, Validation, Writing - review & editing

Conflicts of Interest

The authors declare there is no conflict of interest.

References

[1] M. Dorigo, G. Theraulaz, V. Trianni, "Reflections on the future of swarm robotics," Sci. Robot., vol. 5, no. 49, eabe4385, 2021. https://doi.org/10.1126/scirobot.abe4385

[2] M. Schranz, M. Umlauft, M. Sende, W. Elmenreich, "Swarm robotic behaviors and current applications," Front. Robot. AI, vol. 7, p. 36, 2020. https://doi.org/10.3389/frobt.2020.00036

[3] P. G. F. Dias, M. C. Silva, G. P. Rocha Filho et al., "Swarm robotics: A perspective on the latest reviewed concepts and applications," Sensors, vol. 21, no. 6, p. 2062, 2021. https://doi.org/10.3390/s21062062

[4] E. Debie, K. Kasmarik, M. Garratt, "Swarm robotics: A survey from a multi-tasking perspective," ACM Comput. Surv., vol. 56, no. 2, pp. 1–38, 2024. https://doi.org/10.1145/3610222

[5] K. Azadeh, R. de Koster, D. Roy, "Robotized and automated warehouse systems: Review and recent developments," Transp. Sci., vol. 53, no. 4, pp. 917–945, 2019. https://doi.org/10.1287/trsc.2018.0873

[6] J. P. Queralta, J. Taipalmaa, B. C. Pullinen et al., "Collaborative multi-robot search and rescue," IEEE Access, vol. 8, pp. 191617–191643, 2020. https://doi.org/10.1109/ACCESS.2020.3030163

[7] Y. Alqudsi, M. Makaraci, "Exploring advancements and emerging trends in robotic swarm coordination and control," Proc. Inst. Mech. Eng. C, 2025. https://doi.org/10.1177/09544062251314588

[8] A. Prorok, M. Malencia, L. Carlone et al., "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," arXiv:2109.12343, 2021.

[9] H. Zhang, Y. Chen, Y. Wang et al., "Congestion-aware multi-robot coordination in dynamic environments," Robot. Auton. Syst., vol. 143, p. 103817, 2021. https://doi.org/10.1016/j.robot.2021.103817

[10] G. Yu, M. T. Wolf, "Congestion prediction for large fleets of mobile robots," Proc. IEEE ICRA, pp. 7642–7649, 2023. https://doi.org/10.1109/ICRA48891.2023.10161519

[11] J. van den Berg, M. Lin, D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in Robotics Research, Springer, pp. 3–19, 2019. https://doi.org/10.1007/978-3-319-60916-4_1

[12] J. Snape, J. van den Berg, S. J. Guy, D. Manocha, "Smooth and collision-free navigation for multiple robots using reciprocal velocity obstacles," Auton. Robots, vol. 45, no. 1, pp. 15–30, 2021. https://doi.org/10.1007/s10514-020-09960-7

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots: 35 years of impact," Int. J. Robot. Res., vol. 41, no. 6, pp. 493–499, 2022. https://doi.org/10.1177/02783649221080435

[14] Y. F. Chen, M. Everett, M. Liu, J. P. How, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep RL," Proc. IEEE ICRA, pp. 6015–6022, 2019. https://doi.org/10.1109/ICRA.2019.8794134

[15] P. Long, T. Fan, X. Liao, W. Liu et al., "Towards optimally decentralized multi-robot collision avoidance via deep RL," Int. J. Robot. Res., vol. 39, no. 7, pp. 856–892, 2020. https://doi.org/10.1177/0278364920937741

[16] R. Lowe, Y. Wu, A. Tamar et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in Multi-Agent Machine Learning, Springer, pp. 117–140, 2020. https://doi.org/10.1007/978-3-031-18077-5_6

[17] C. Yu, A. Velu, E. Vinitsky et al., "The surprising effectiveness of PPO in cooperative multi-agent games," Adv. Neural Inf. Process. Syst., vol. 35, 2022. arXiv: 2103.01955

[18] R. Stern, N. Sturtevant, A. Felner et al., "Multi-agent pathfinding: Definitions, variants, and benchmarks," Artif. Intell., vol. 303, p. 103650, 2022. https://doi.org/10.1016/j.artint.2022.103650

[19] G. Sartoretti, J. Kerr, Y. Shi et al., "PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning," IEEE Robot. Autom. Lett., vol. 4, no. 3, pp. 2378–2385, 2019. https://doi.org/10.1109/LRA.2019.2903261

[20] A. Andreychuk, K. Yakovlev, E. Boyarski, R. Stern, "Multi-agent pathfinding with continuous time," Artif. Intell., vol. 305, p. 103662, 2022. https://doi.org/10.1016/j.artint.2022.103662

[21] J. Gao, Y. Li, X. Li et al., "A review of graph-based multi-agent pathfinding solvers," Knowl.-Based Syst., vol. 283, p. 111121, 2024. https://doi.org/10.1016/j.knosys.2023.111121

[22] J. Li, W. Ruml, S. Koenig, "EECBS: A bounded-suboptimal search for multi-agent path finding," Proc. AAAI, vol. 35, no. 14, pp. 12353–12362, 2021. https://doi.org/10.1609/aaai.v35i14.17466

[23] W. Hönig, J. A. Preiss, T. K. S. Kumar et al., "Trajectory planning for quadrotor swarms," IEEE Trans. Robot., vol. 38, no. 2, pp. 1083–1100, 2022. https://doi.org/10.1109/TRO.2021.3098745

[24] L. Wang, A. D. Ames, M. Egerstedt, "Safety barrier certificates for collision-free multirobot systems," IEEE Trans. Robot., vol. 33, no. 3, pp. 661–674, 2021. https://doi.org/10.1109/TRO.2017.2659727

[25] T. Fan, P. Long, W. Liu, J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," Int. J. Robot. Res., vol. 39, no. 7, pp. 856–892, 2020. https://doi.org/10.1177/0278364920916531

[26] M. Everett, Y. F. Chen, J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," IEEE Access, vol. 9, pp. 10357–10377, 2021. https://doi.org/10.1109/ACCESS.2021.3050338

[27] D. Zhou, Z. Wang, M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using buffered Voronoi cells," Proc. IEEE ICRA, 2020. https://doi.org/10.1109/ICRA40945.2020.9196591

[28] A. Pierson, W. Schwarting, S. Karaman, D. Rus, "Weighted buffered Voronoi cells for distributed semi-cooperative behavior," Proc. IEEE ICRA, pp. 5611–5617, 2020. https://doi.org/10.1109/ICRA40945.2020.9197589

[29] Q. Li, F. Gama, A. Ribeiro, A. Prorok, "Graph neural networks for decentralized multi-robot path planning," Proc. IEEE/RSJ IROS, pp. 11785–11792, 2020. https://doi.org/10.1109/IROS45743.2020.9341668

[30] C. Jiang, X. Huang, Y. Guo, "End-to-end decentralized formation control using a GNN-based learning method," Front. Robot. AI, vol. 10, p. 1285412, 2023. https://doi.org/10.3389/frobt.2023.1285412

[31] M. Damani, Z. Luo, E. Wenzel, G. Sartoretti, "PRIMAL2: Pathfinding via reinforcement and imitation multi-agent learning – lifelong," IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 2666–2673, 2021. https://doi.org/10.1109/LRA.2021.3062365

[32] J. Wang, Z. Ren, T. Liu et al., "QPLEX: Duplex dueling multi-agent Q-learning," Adv. Neural Inf. Process. Syst., vol. 33, pp. 19934–19945, 2020. arXiv: 2008.01062

[33] J. Orr, A. Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: A survey," Sensors, vol. 23, no. 7, p. 3625, 2023. https://doi.org/10.3390/s23073625

[34] W. Zhao, J. P. Queralta, T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," Proc. IEEE Symp. Ser. Comput. Intell., pp. 737–744, 2020. https://doi.org/10.1109/SSCI47803.2020.9308468

[35] C. Amato, "An introduction to centralized training for decentralized execution in cooperative multi-agent RL," arXiv: 2409.03052, 2024.

[36] Y. Zhou, S. Liu, Y. Qing et al., "Is centralized training with decentralized execution framework centralized enough for MARL?" Proc. IJCAI, 2024. https://doi.org/10.24963/ijcai.2024/14

[37] J. Blumenkamp, S. Morad, J. Gielis, Q. Li, A. Prorok, "A framework for real-world multi-robot systems running decentralized GNN-based policies," Proc. IEEE ICRA, pp. 8772–8778, 2022. https://doi.org/10.1109/ICRA46639.2022.9812330

[38] D. Domini, F. Cavallari, G. Aguzzi, M. Viroli, "Scaling swarm coordination with GNNs – how far can we go?" AI, vol. 6, no. 11, p. 282, 2025. https://doi.org/10.3390/ai6110282

[39] M. Goarin, G. Loianno, "Graph neural network for decentralized multi-robot goal assignment," IEEE Robot. Autom. Lett., vol. 9, no. 4, pp. 3456–3463, 2024. https://doi.org/10.1109/LRA.2024.3367040

[40] A. Cheraghi, S. Shahzad, K. Graffi, "Past, present, and future of swarm robotics," in Intelligent Systems and Applications (IntelliSys), Springer, pp. 190–233, 2022. https://doi.org/10.1007/978-3-031-16078-3_14