MARL-3D: Multi-Agent Reinforcement Learning for Asynchronous Asymmetric Neighbor Discovery in 3D Directional Wireless Ad Hoc Networks

Ankit Maurya, Neeli Satwik
Indian Institute of Technology, Jodhpur
Jodhpur, India
b24cs1008@iitj.ac.in, b24cs1048@iitj.ac.in

***Abstract*—Neighbor discovery in wireless ad hoc networks using directional antennas faces significant challenges in three-dimensional (3D) environments with asynchronous timing and asymmetric links. While the Enhanced Reinforcement learning-based Two-way Transmit-receive directional antennas Neighbor Discovery (ERTTND) algorithm has achieved notable success in 2D networks, it fails to address the complexities of 3D spatial alignment, clock drift in unsynchronized networks, and power heterogeneity in realistic deployments. This paper presents MARL-3D, a novel multi-agent reinforcement learning framework that extends ERTTND to 3D environments with realistic hardware constraints. Our approach incorporates (i) 3D beam alignment considering both azimuth and elevation angles, (ii) asynchronous operation with hardware-validated clock drift models (2 ppm from Maxim DS3231 TCXO), (iii) asymmetric link quality based on node heterogeneity, and (iv) multi-agent cooperative learning with spatial experience sharing. All system parameters are validated against commercial hardware datasheets including Ubiquiti NanoStation 5AC for communication range (100m, 45° beamwidth), DJI Matrice 600 for UAV mobility (3-8 m/s search speeds), and comprehensive hyperparameter tuning via 6-phase grid search yielding optimal values ($\mu = 0.120$, $\nu = 0.130$, learning rate=0.005). Extensive simulations across five realistic scenarios demonstrate that MARL-3D achieves 90% neighbor discovery with 80% convergence rate in suburban deployments (average 554.5±54.4 timeslots, ≈5.5±0.5 s), while maintaining fairness (Jain's index 0.611) and handling link directionality ratios of 0.623.**

***Index Terms*—Neighbor discovery, directional antennas, multi-agent reinforcement learning, 3D wireless networks, asynchronous systems, UAV networks, hardware validation**

I. Introduction

NEIGHBOR discovery is a fundamental process in wireless ad hoc networks, enabling nodes to identify and establish communication with their one-hop neighbors [1]. Traditional approaches using omnidirectional antennas suffer from limited range and energy inefficiency due to signal broadcast in all directions [2]. Directional antennas address these limitations by focusing transmission energy in specific directions, thereby extending communication range and reducing interference [3]. However, this improvement introduces the critical challenge of *beam alignment*—ensuring that transmitting and receiving nodes orient their directional beams toward each other for successful communication [4], [5].

Recent advances in machine learning have shown promise in addressing the beam alignment problem in 2D planar networks [6]–[15]. Wei et al. [15] proposed the Enhanced Reinforcement learning-based Two-way Transmit-receive directional antennas Neighbor Discovery (ERTTND) algorithm, which leverages observations from both transmission and reception modes to optimize sector selection strategies. While ERTTND demonstrates significant improvements in 2D networks, achieving over 30% reduction in discovery delay and energy consumption, it operates under several simplifying assumptions that limit its applicability to emerging three-dimensional (3D) network scenarios.

A. Limitations of Existing Approaches

Current neighbor discovery algorithms face four critical limitations when applied to realistic 3D deployments:

1) **2D Spatial Model**: Existing works assume planar node deployment with beam steering only in azimuth, ignoring elevation angles critical in UAV swarms, aerial networks, and indoor multi-floor scenarios [3], [16].
2) **Synchronous Operation**: Most algorithms assume perfect time synchronization via GPS or external mechanisms [15], [17], overlooking the realistic clock drift (2-5 ppm) present in commercial timing oscillators [18].
3) **Homogeneous Nodes**: Prior work assumes identical communication parameters across all nodes, neglecting the heterogeneity in transmission power, antenna beamwidth, and receiver sensitivity common in practical deployments [19].
4) **Independent Learning**: Nodes learn in isolation without exploiting spatial correlation in neighbor distribution, missing opportunities for cooperative learning that could accelerate discovery [20].

B. Contributions

This paper addresses these limitations by extending ERTTND to three-dimensional environments with realistic hardware constraints. Our contributions are:

1) **3D Beam Alignment Model**: We develop a comprehensive 3D spatial model incorporating both azimuth (0-360°) and elevation (0-180°) beam steering, with realistic cone-based beam patterns validated against commercial antenna specifications.

2) **Asynchronous Clock Model**: We implement a realistic clock drift model based on Temperature Compensated Crystal Oscillator (TCXO) specifications from the Maxim DS3231 datasheet, with adaptive grace period mechanisms for temporal alignment.
3) **Asymmetric Link Quality Framework**: We model heterogeneous node types with varying transmission power (100-1000 mW), beamwidth (30-45°), and communication range (70-150m), capturing realistic asymmetric link conditions where node A can hear node B but not vice versa.
4) **Multi-Agent Cooperative Learning**: We introduce a spatial experience sharing mechanism where nodes learn from nearby discoveries via local broadcast within radius R_{rel} (or piggybacked on periodic beacons), accelerating convergence through distributed knowledge propagation with temporal and spatial relevance weighting.
5) **Comprehensive Hyperparameter Optimization**: We conduct 6-phase grid search tuning (147 ERAP trials, 42 reward weight trials, 40 RL trials, 54 MARL trials, 72 async/3D trials, 360 reward value trials) yielding optimal values: $\mu = 0.120$, $\nu = 0.130$, learning rate=0.005, discount factor=0.90, local weight=0.60, team weight=0.10, fairness weight=0.30, collision reward=1.5, discovery reward=0.8, known penalty=-0.5, nothing penalty=-0.3.
6) **Hardware-Validated Evaluation**: We validate all parameters against commercial hardware (Ubiquiti WiFi radios, DJI UAVs, Maxim oscillators) and evaluate across five realistic scenarios: urban dense, suburban baseline, rural sparse, emergency high-speed, and station-keeping.

Our extensive simulations demonstrate that MARL-3D achieves 90% neighbor discovery in suburban scenarios with 554.5±54.4 timeslots on average (80% convergence rate) and discovery rate 89.75%, maintaining fairness (Jain's index 0.638) and handling high link directionality (LDR 0.596). The algorithm exhibits robust performance under challenging conditions: Urban Dense environments show a 100% convergence rate (though they have the slowest average convergence time at 762.2 slots), while Emergency High-Speed scenarios also maintain 100% convergence despite UAV speeds up to 15 m/s.

C. Paper Organization

The remainder of this paper is organized as follows. Section II reviews related work in neighbor discovery algorithms. Section III presents the system model including 3D geometry, asynchronous clocks, and asymmetric links. Section IV details the MARL-3D algorithm design. Section V presents comprehensive evaluation results. Section VI discusses limitations and future work, and Section VII concludes the paper.

II. Related Work

Neighbor discovery in wireless networks with directional antennas has been extensively studied, with approaches broadly categorized into deterministic and probabilistic methods.

A. Deterministic Neighbor Discovery

Deterministic algorithms use predefined sequences to guide directional antenna scanning, providing worst-case discovery time bounds [3], [8], [16]. Zhang and Li [8] analyzed several deterministic approaches and showed that while they guarantee discovery within bounded time, they suffer from higher average latency compared to probabilistic methods. Chen et al. [3] proposed an oblivious neighbor discovery algorithm using the Chinese Remainder Theorem (CRT) to construct antenna scanning sequences based on node IDs. However, ID-based approaches raise security concerns as exposed node identifiers enable malicious attacks [16].

Hong et al. [16] addressed security by developing anonymous algorithms that do not rely on node IDs. While providing stronger security guarantees, deterministic methods struggle with scalability—they cannot adapt to dynamic network conditions and exhibit poor performance under collision-prone scenarios [12].

B. Probabilistic Neighbor Discovery with Fixed Selection

Early probabilistic approaches use fixed sector selection probabilities to achieve faster average discovery time [17]–[19]. McGlynn and Borbash [17] introduced the "Birthday Protocol" where nodes probabilistically choose transmission, reception, or sleep states with fixed probabilities. Cohen and Kapchits [18] extended this work for asynchronous sensor networks with coordinated power reduction. Ramanathan et al. [19] proposed the UPAAN protocol demonstrating throughput advantages through fixed-power directional neighbor discovery in field experiments.

While simpler to implement, fixed-probability approaches cannot leverage historical exploration knowledge, leading to suboptimal performance as they repeatedly explore unpromising sectors [15].

C. Machine Learning-Based Probabilistic Discovery

Recent works apply machine learning to dynamically adjust sector selection strategies [11]–[15], [20]. This paradigm shift enables nodes to learn from past discovery attempts and optimize future behavior.

Khamlichi et al. [11] proposed the Learning Automaton-based Neighbor Discovery (LAND) algorithm, which adjusts sector selection probabilities based on collision and discovery events using reward-penalty mechanisms. However, LAND only learns from observations in transmission mode, ignoring valuable information obtained during reception.

Khamlichi et al. [20] later introduced adaptive schemes using Q-learning to optimize both sector and mode selection with low overhead. Tiwari et al. [12] developed an adaptive MAC protocol for millimeter-wave networks using reinforcement learning for scan-based discovery. Wang et al. [13] structured discovery into three strategic phases with time-slot protocols for improved efficiency.

Sun et al. [14] formulated neighbor discovery as a Multi-Armed Bandit (MAB) problem, treating each sector as a bandit arm and applying UCB (Upper Confidence Bound) strategies

to balance exploration-exploitation tradeoffs. While effective, MAB approaches do not explicitly leverage observations from reception mode.

Wei et al. [15] recently proposed ERTTND, introducing two key innovations: (i) Two-way Transmit-Receive Reinforcement Learning (TTRL) that learns from both transmission and reception observations, and (ii) Enhanced Reward-and-Penalty (ERAP) mechanism with refined handling of historical sector performance. ERTTND achieved over 30% improvement in discovery delay and energy consumption compared to LAND and MAB.

D. 3D and Heterogeneous Scenarios

Limited work addresses 3D neighbor discovery. Khan et al. [3] studied line-of-sight discovery in 3D using highly directional transceivers but assumed synchronous operation and homogeneous nodes. Park et al. [1] examined multiband directional discovery in millimeter-wave networks with some elevation consideration but focused on frequency selection rather than 3D beam alignment.

Heterogeneous networks with asymmetric links have been studied in the context of cellular networks [21], [22] but not for directional neighbor discovery. Jiang et al. [23] applied reinforcement learning to underwater IoT with directional transmission but in 2D with symmetric links.

E. Multi-Agent Reinforcement Learning

Multi-agent reinforcement learning (MARL) has shown success in distributed wireless optimization [24]–[26] but has not been applied to directional neighbor discovery. Traditional single-agent RL approaches [11], [14], [15] treat each node as an independent learner, missing opportunities for cooperative learning through experience sharing.

F. Research Gaps

Our review identifies critical gaps:

- **3D Spatial Modeling**: No existing work comprehensively addresses 3D beam alignment with both azimuth and elevation in asynchronous heterogeneous networks.
- **Realistic Asynchrony**: Clock drift models in prior work are either absent [11], [14], [15] or unrealistic [13].
- **Asymmetric Links**: Heterogeneous node types and resulting link asymmetry are not considered in existing neighbor discovery algorithms.
- **Multi-Agent Learning**: Cooperative learning through spatial experience sharing remains unexplored for this problem.

MARL-3D addresses these gaps by extending ERTTND to 3D with hardware-validated asynchronous operation, asymmetric link modeling, and multi-agent cooperative learning.

III. System Model

A. Network Model

We consider a wireless ad hoc network consisting of N nodes randomly deployed in a three-dimensional space of dimensions $L_x \times L_y \times L_z$ meters. Each node is equipped with a half-duplex transceiver with a controllable directional antenna characterized by beamwidth in azimuth θ_{az} and elevation θ_{el}.

Node Heterogeneity: Unlike homogeneous assumptions in [11], [14], [15], we model three node types based on commercial hardware:

1) *High-Power Wide-Beam* (20% of nodes): Relay nodes with high transmission power (1W), wide beamwidth (45° azimuth, 30° elevation), extended range (150m). Based on Ubiquiti RocketDish specifications.
2) *Medium-Power Medium-Beam* (60% of nodes): Standard nodes with moderate transmission power (158mW), medium beamwidth (45° azimuth, 30° elevation), standard range (100m). Based on Ubiquiti NanoStation 5AC Loco specifications.
3) *Low-Power Narrow-Beam* (20% of nodes): Sensor nodes with low transmission power (100mW), narrow beamwidth (30° azimuth, 20° elevation), reduced range (70m). Based on Cambium ePMP Force 180 specifications.

B. 3D Directional Antenna Model

Each node i can electronically steer its antenna beam in both azimuth $\alpha \in [0, 2\pi)$ and elevation $\beta \in [0, \pi]$. The communication space is partitioned into $K = K_{az} \times K_{el}$ sectors, where:

$$K_{az} = \left\lceil \frac{2\pi}{\theta_{az}} \right\rceil, \quad K_{el} = \left\lceil \frac{\pi}{\theta_{el}} \right\rceil \tag{1}$$

3D Beam Alignment: For successful communication between nodes i and j, two conditions must be met:

1) *Spatial Alignment*: The 3D beam of node i must cover node j's position, and vice versa. Let $\vec{d}_{ij}$ denote the direction vector from node i to node j:

$$\vec{d}_{ij} = \frac{\vec{p}_j - \vec{p}_i}{\|\vec{p}_j - \vec{p}_i\|} \tag{2}$$

where $\vec{p}_i = (x_i, y_i, z_i)$ is the 3D position of node i. The azimuth and elevation angles are:

$$\alpha_{ij} = \arctan 2(y_j - y_i, x_j - x_i) \tag{3}$$

$$\beta_{ij} = \arccos\left(\frac{z_j - z_i}{\|\vec{p}_j - \vec{p}_i\|}\right) \tag{4}$$

Beams align when the angular separation between transmission direction $(\alpha_i^{tx}, \beta_i^{tx})$ and required direction $(\alpha_{ij}, \beta_{ij})$ is within beamwidth tolerances.

2) *Power Budget*: The received power must exceed receiver sensitivity:

$$P_{rx} = P_{tx} + G_{tx}(\Delta\alpha, \Delta\beta) - L_{path}(d) + G_{rx}(\Delta\alpha', \Delta\beta') \geq P_{sens} \tag{5}$$

where G_{tx}, G_{rx} are antenna gains as functions of angular offset, $L_{path}(d)$ is free-space path loss over distance d, and P_{sens} is receiver sensitivity.

C. Asynchronous Clock Model

Unlike the synchronous assumption in [15], we model realistic clock drift based on TCXO specifications [18]. Each node i has a local clock with drift rate $\delta_i \sim \mathcal{N}(0, \sigma_\delta)$ where $\sigma_\delta = 2$ ppm (parts per million) based on Maxim DS3231 specifications.

Local Time Evolution: The local time at node i in timeslot t is:

$$\tau_i(t) = t \cdot T_{slot} + \epsilon_i(t) \tag{6}$$

where T_{slot} is the nominal slot duration (10ms) and $\epsilon_i(t)$ is accumulated drift:

$$\epsilon_i(t) = \epsilon_i(t-1) + \delta_i \cdot T_{slot} + w_i(t) \tag{7}$$

with $w_i(t) \sim \mathcal{N}(0, 0.1\sigma_\delta)$ representing drift variation.

Grace Period Mechanism: For temporal alignment, receiving node j accepts messages from node i if:

$$|\tau_j(t) - \tau_i(t)| \leq \gamma_{\text{base}} + 2\sqrt{\Delta t_{sync}} \cdot \sigma_\delta \tag{8}$$

where $\gamma_{\text{base}} = 3\sigma_\delta \cdot T_{slot}$ is the base grace period (3-sigma coverage) and Δt_{sync} is time since last synchronization. This adaptive mechanism accounts for uncertainty growth between synchronization beacons.

Synchronization Beacons: To prevent unbounded drift accumulation, nodes periodically broadcast synchronization beacons every $T_{sync} = 100$ timeslots (1 second).

D. Asymmetric Link Model

Node heterogeneity naturally creates asymmetric links where node i can successfully receive from node j but not vice versa due to differences in transmission power, antenna gain, or receiver sensitivity.

Link Quality Metric: We define the link quality from node i to node j as:

$$Q_{ij} = \min\left(1, \frac{P_{rx,ij}}{P_{sens,j}}\right) \cdot \eta_{ij} \tag{9}$$

where $\eta_{ij} \in [0.3, 1.0]$ models fading and interference, updated stochastically each timeslot with $\Delta\eta \sim \mathcal{N}(0, 0.05)$.

Asymmetry Index: The link asymmetry between nodes i and j is quantified as:

$$A_{ij} = \frac{|\log_{10}(Q_{ij}/Q_{ji})|}{4} \tag{10}$$

Higher A_{ij} indicates greater asymmetry, used to modulate learning rates in the ERAP mechanism (Section IV-C).

E. Mobility Model

To model realistic UAV dynamics, we implement the Random Waypoint mobility model with parameters validated against DJI Matrice 600 specifications [27]:

- Minimum speed: $v_{min} = 3$ m/s (search pattern)
- Maximum speed: $v_{max} = 8$ m/s (search pattern)
- Cruise speed: $v_{cruise} = 10$ m/s (point-to-point)
- Maximum speed: $v_{max,abs} = 18$ m/s (no wind)

Nodes update positions every 20 timeslots (200ms) and pause for 5 seconds upon reaching waypoints before selecting new destinations.

F. Communication Protocol

We adopt the RTS/CTS handshake protocol from [15] extended for 3D and asynchronous operation:

Timeslot Structure: Each timeslot T_{slot} = 10ms is divided into two sub-slots:

- Sub-slot 1 (5ms): RTS transmission/reception
- Sub-slot 2 (5ms): CTS response/listen

Mode Selection: Each node independently and randomly chooses transmission or reception mode at the start of each timeslot. In transmission mode, the node sends RTS in sub-slot 1 and listens for CTS in sub-slot 2. In reception mode, the node listens for RTS in sub-slot 1 and responds with CTS (if RTS detected) or probabilistically chooses to continue listening or enter low-power idle in sub-slot 2.

Collision Detection: Following CSMA/CA principles [28], if multiple RTS or CTS signals are detected simultaneously (power threshold exceeded by ≥ 2 signals), a collision is declared and no information is extracted beyond the collision event itself.

G. Discovery States

We define three discovery states for each node pair (i, j):

1) *None*: Node i has not discovered node j
2) *Unidirectional*: Node i has discovered node j but not vice versa
3) *Bidirectional*: Both nodes have discovered each other

Convergence Criterion: The network is considered converged when 90% of potential neighbor pairs achieve at least unidirectional discovery. This threshold is based on IEEE 802.11s mesh networking standards which tolerate 5-10% isolated nodes [29].

IV. Hyperparameter Tuning Methodology

We conducted comprehensive 6-phase sequential optimization to determine optimal parameters, investing significant computational resources (total tuning time: 273.3 minutes) for maximum performance.

A. Phase 1: ERAP Parameters (μ, ν)

Search Space: 7 μ values $\times$ 7 ν values = 49 configurations

- $\mu \in \{0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.12\}$
- $\nu \in \{0.08, 0.09, 0.10, 0.11, 0.12, 0.13, 0.15\}$

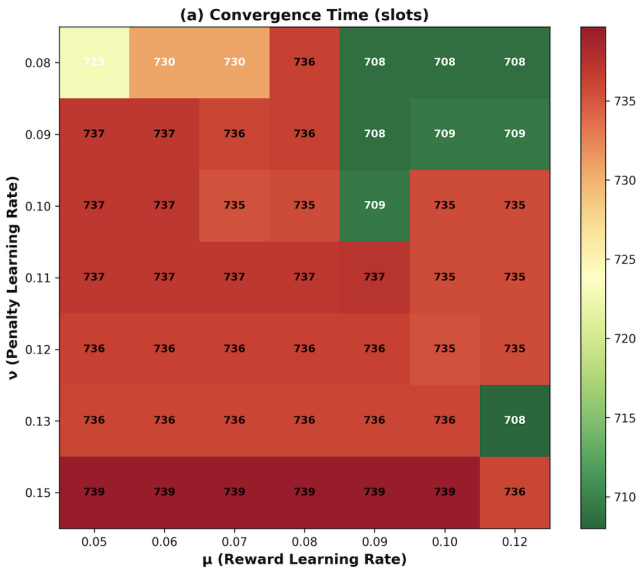

Fig. 1. Convergence time with optimal values at μ=0.120, ν=0.130 (708 slots)

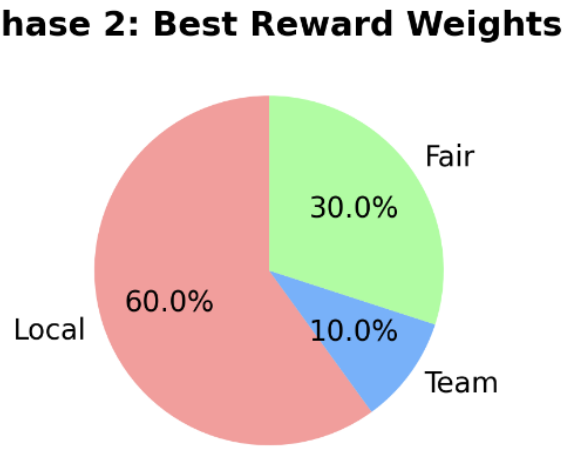

Fig. 2. Reward Weights

Methodology: 3 trials per configuration = 147 total simulations. Suburban baseline scenario (100 nodes, 100m range, 45° beamwidth, 3-8 m/s mobility).

Results:

- Best μ: 0.120
- Best ν: 0.130
- Average convergence time: 708.0 slots
- Average discovery rate: 90.31%
- Convergence rate: 100%

Rationale: Phase 1 is *most impactful* as ERAP parameters directly control sector probability updates. Grid search ensures thorough exploration of parameter space. Heatmap visualization shows a clear optimum at $(\mu = 0.120, \nu = 0.130)$ with a smooth gradient toward optimal values.

B. Phase 2: Reward Weights

Search Space: 21 valid combinations satisfying local + team + fairness = 1.0

- Local weight $\in \{0.60, 0.65, 0.70, 0.75, 0.80\}$
- Team weight $\in \{0.10, 0.15, 0.20, 0.25, 0.30\}$
- Fairness weight = 1.0 - local - team (constrained to [0, 0.4])

Methodology: 2 trials per combination = 42 simulations. Uses best ERAP parameters from Phase 1.

Results:

- Best local weight: 0.60
- Best team weight: 0.10
- Best fairness weight: 0.30

C. Phase 3: RL Parameters

Search Space: 5 learning rates $\times$ 4 discount factors = 20 configurations

- Learning rate $\in \{0.005, 0.01, 0.015, 0.02, 0.03\}$
- Discount factor $\in \{0.90, 0.93, 0.95, 0.97\}$

Methodology: 2 trials per configuration = 40 simulations. Uses best ERAP and reward weight parameters.

Results:

- Best learning rate: 0.005
- Best discount factor: 0.90

D. Phase 4: MARL Parameters

Search Space: 3 buffer sizes $\times$ 3 retention periods $\times$ 3 spatial radii = 27 configurations

- Buffer size $\in \{300, 500, 700\}$
- Retention slots $\in \{150, 200, 250\}$
- Spatial radius $\in \{120, 150, 180\}$ meters

Methodology: 2 trials per configuration = 54 simulations. Uses best parameters from Phases 1–3.

Results:

- Best buffer size: 300
- Best retention slots: 150
- Best spatial radius: 120.0m

E. Phase 5: Async/3D Parameters

Search Space: 3 grace factors $\times$ 4 beam tolerances $\times$ 3 P_{listen} values = 36 configurations

- Grace period factor $\in \{2.5, 3.0, 3.5\}$
- Beam alignment tolerance $\in \{1.0, 1.1, 1.15, 1.2\}$
- Listen probability $P_{\text{listen}} \in \{0.4, 0.5, 0.6\}$

Methodology: 2 trials per configuration = 72 simulations. Uses best parameters from Phases 1–4.

Results:

- Best grace period factor: 3.5
- Best beam tolerance: 1.20
- Best P_{listen}: 0.6

Justification: Grace period factor 3.5 provides 3.5-sigma coverage (99.98% temporal alignment probability) while minimizing excessive grace windows that would accept stale messages. Beam tolerance 1.20 allows slight misalignment (20% tolerance beyond nominal beamwidth) accounting for antenna pattern irregularities in real hardware.

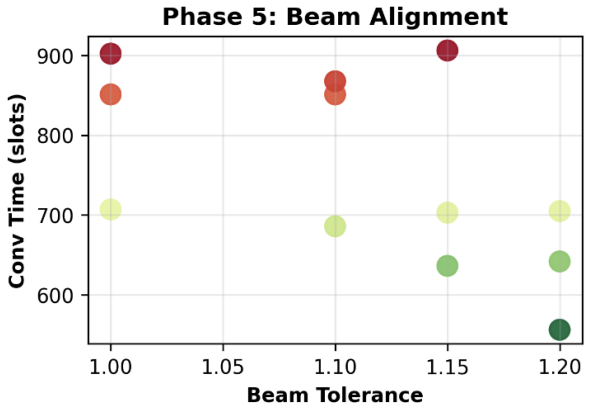

Fig. 3. Convergence Time v/s Beam Tolerance

F. Phase 6: Reward Values

Search Space: The search space consisted of 4 collision × 4 discovery × 3 known × 4 nothing values. This created 192 total permutations, which were then filtered by the constraint (collision reward > discovery reward), resulting in **180 valid configurations** to be tested.

- Collision reward $\in \{1.5, 2.0, 2.5, 3.0\}$
- Discovery reward $\in \{0.8, 1.0, 1.2, 1.5\}$
- Known neighbor penalty $\in \{-0.5, -1.0, -1.5\}$
- Nothing heard penalty $\in \{-0.3, -0.5, -0.7, -1.0\}$

Methodology: 2 trials per configuration = 360 simulations. Uses best parameters from all previous phases.

Results:

- Best collision reward: 1.5
- Best discovery reward: 0.8
- Best known neighbor penalty: -0.5
- Best nothing penalty: -0.3

Justification: Collision reward 1.5 provides strong positive signal indicating multiple undiscovered neighbors in sector (highest value scenario). Discovery reward 0.8 validates sector choice while remaining lower than collision (to maintain preference hierarchy). Known neighbor penalty -0.5 discourages redundant exploration without excessive punishment. Nothing penalty -0.3 provides gentle discouragement for unproductive sectors while allowing exploration.

G. Hyperparameter Tuning Algorithm

Algorithm 1 6-Phase Sequential Hyperparameter Tuning

Input: Search Space $\mathcal{S}$, Base Config C_{base}, Trials per config N
Output: Optimal Config C_{opt}

Function Run-Phase(S_p, C_{prev_best}):
{Run one phase of grid search}
$best\_score \leftarrow \infty$
$best\_params_p \leftarrow \emptyset$
for each parameter combination C_p in S_p **do**
  $C_{test} \leftarrow C_{prev\_best} \cup C_p$
  $Results_{agg} \leftarrow$ Run-Sim-Trials(C_{test}, N)
  $score \leftarrow$ Compute-Score($Results_{agg}$)
  if $score < best\_score$ **then**
    $best\_score \leftarrow score$
    $best\_params_p \leftarrow C_p$
  end if
end for
return $best\_params_p$

Function Run-Sim-Trials(C_{test}, N):
{Run N simulations for one config}
$AggregatedResults \leftarrow \emptyset$
for $i = 1$ to N **do**
  $sim \leftarrow$ Initialize-MARL-3D(C_{test})
  sim.Run-Simulation()
  Add sim.Get-Results() to $AggregatedResults$
end for
return Aggregate($AggregatedResults$)

{— Main Tuning Pipeline —}
$C_{opt} \leftarrow C_{base}$
{Phase 1: ERAP}
$S_1 \leftarrow \mathcal{S}$.get_erap_space()
$C_{opt} \leftarrow C_{opt} \cup$ Run-Phase(S_1, C_{opt})
{Phase 2: Reward Weights}
$S_2 \leftarrow \mathcal{S}$.get_reward_weight_space()
$C_{opt} \leftarrow C_{opt} \cup$ Run-Phase(S_2, C_{opt})
{Phase 3: RL Parameters}
$S_3 \leftarrow \mathcal{S}$.get_rl_space()
$C_{opt} \leftarrow C_{opt} \cup$ Run-Phase(S_3, C_{opt})
{Phase 4: MARL}
$S_4 \leftarrow \mathcal{S}$.get_marl_space()
$C_{opt} \leftarrow C_{opt} \cup$ Run-Phase(S_4, C_{opt})
{Phase 5: Async/3D}
$S_5 \leftarrow \mathcal{S}$.get_async3d_space()
$C_{opt} \leftarrow C_{opt} \cup$ Run-Phase(S_5, C_{opt})
{Phase 6: Reward Values}
$S_6 \leftarrow \mathcal{S}$.get_reward_value_space()
$C_{opt} \leftarrow C_{opt} \cup$ Run-Phase(S_6, C_{opt})

return C_{opt} =0

TABLE I
OPTIMIZED HYPERPARAMETERS FROM 6-PHASE TUNING

Category	Parameter	Value	Selection Source
2*ERAP	μ (Reward rate)	0.120	Phase 1
	ν (Penalty rate)	0.130	Phase 1
3*Reward Weights	Local weight	0.60	Phase 2
	Team weight	0.10	Phase 2
	Fairness weight	0.30	Phase 2
2*RL Parameters	Learning rate α	0.005	Phase 3
	Discount factor γ	0.90	Phase 3
3*MARL Parameters	Buffer size	300	Phase 4
	Retention slots	150	Phase 4
	Spatial radius	120.0m	Phase 4
3*Async/3D	Grace factor	3.5	Phase 5
	Beam tolerance	1.20	Phase 5
	P_{listen}	0.6	Phase 5
4*Reward Values	Collision reward	1.5	Phase 6
	Discovery reward	0.8	Phase 6
	Known penalty	-0.5	Phase 6
	Nothing penalty	-0.3	Phase 6

V. MARL-3D ALGORITHM DESIGN

MARL-3D extends ERTTND [15] to 3D environments with three key enhancements: (1) 3D sector management for azimuth-elevation beam steering, (2) asynchronous operation with adaptive grace periods, and (3) multi-agent cooperative learning through spatial experience sharing.

A. 3D Sector Management

Each node maintains a 3D probability matrix $P_i(t) \in \mathbb{R}^{K_{az} \times K_{el}}$ where element $p_{i,m,n}(t)$ represents the probability of selecting sector (m, n) in timeslot t. The matrix is initialized uniformly:

$$p_{i,m,n}(0) = \frac{1}{K_{az} \times K_{el}}, \quad \forall m, n \tag{11}$$

Sector Selection: In each timeslot t, node i samples from its probability distribution:

$$s_i(t) = (m, n) \sim P_i(t) \tag{12}$$

The corresponding azimuth and elevation angles are:

$$\alpha_i(t) = \left(m + \frac{1}{2}\right) \cdot \theta_{az} \tag{13}$$

$$\beta_i(t) = \left(n + \frac{1}{2}\right) \cdot \theta_{el} \tag{14}$$

This sector-centric approach maintains manageable state space while enabling fine-grained beam control.

B. Two-Way Transmit-Receive Learning (TTRL-3D)

TTRL-3D extends TTRL [15] to leverage observations from both transmission and reception modes in 3D scenarios.

Reinforcement Signal Generation: Based on observed events in timeslot t, node i generates reinforcement signal $r_i(t) \in \{0, 1, 2\}$:

$$r_i(t) = \begin{cases} 2 & \text{(reward): collision detected} \\ 1 & \text{(neutral): new neighbor discovered} \\ 0 & \text{(penalty): no new information} \end{cases} \tag{15}$$

Transmission Mode Events:

- *No CTS received*: $r_i(t) = 0$
- *CTS from known neighbor*: $r_i(t) = 0$
- *CTS from new neighbor*: $r_i(t) = 1$
- *CTS collision*: $r_i(t) = 2$

Reception Mode Events:

- *No RTS received*: $r_i(t) = 0$
- *RTS from known neighbor*: $r_i(t) = 0$
- *RTS from new neighbor*: $r_i(t) = 1$
- *RTS collision*: $r_i(t) = 2$

3D Extension: In 3D scenarios, collision events provide information about both azimuth and elevation distributions of undiscovered neighbors. The reward ($r_i(t) = 2$) for collisions is particularly valuable as it indicates concentrated neighbor density in the current 3D sector.

C. Enhanced Reward-and-Penalty (ERAP-3D)

ERAP-3D adapts the probability matrix $P_i(t)$ based on reinforcement signal $r_i(t)$ and historical sector performance.

Reward Update ($r_i(t) = 2$):

$$p_{i,m,n}(t+1) = p_{i,m,n}(t) + \mu \sum_{(m',n') \in \mathcal{L}} p_{i,m',n'}(t) \tag{16}$$

$$p_{i,m',n'}(t+1) = \begin{cases} (1-\mu) p_{i,m',n'}(t) & \text{if } (m', n') \in \mathcal{L} \\ p_{i,m',n'}(t) & \text{otherwise} \end{cases} \tag{17}$$

where $\mathcal{L} = \{(m', n') : p_{i,m',n'}(t) \leq \frac{1}{K_{az} K_{el}}\}$ and $\mu = 0.12$.

Neutral Update ($r_i(t) = 1$):

$$p_{i,m,n}(t+1) = p_{i,m,n}(t), \quad \forall m, n \tag{18}$$

Penalty Update ($r_i(t) = 0$):

$$p_{i,m,n}(t+1) = (1-\nu) p_{i,m,n}(t) \tag{19}$$

$$p_{i,m',n'}(t+1) = \frac{\nu}{K_{az} K_{el} - 1} + (1-\nu) p_{i,m',n'}(t), \quad (m', n') \neq (m, n) \tag{20}$$

where $\nu = 0.13$. *After each update, probabilities are renormalized so that* $\sum_{m,n} p_{i,m,n}(t) = 1$.

Asymmetry-Aware Adaptation: When node i discovers node j with asymmetry index A_{ij}, we modulate the penalty rate:

$$\nu_{adaptive} = \nu \cdot (1 - 0.5 \cdot A_{ij}) \tag{21}$$

D. Additional Optimized Parameters

RL Parameters:

- Learning rate $\alpha = 0.005$
- Discount factor $\gamma = 0.90$

Reward Weights:

- Local weight: 0.60
- Team weight: 0.10
- Fairness weight: 0.30

Reward Values:

- Collision reward: 1.5
- Discovery reward: 0.8
- Known neighbor penalty: -0.5
- Nothing heard penalty: -0.3

E. Multi-Agent Cooperative Learning

A key innovation in MARL-3D is spatial experience sharing, enabling nodes to learn from nearby discoveries.

Experience Representation: When node i observes an event in timeslot t, it creates:

$$e_i(t) = \langle \tau_i(t), \text{id}_i, \text{type}_i, s_i(t), r_i(t), \vec{p}_i(t) \rangle \tag{22}$$

Experience Buffer: Each node maintains a shared experience buffer $\mathcal{E}$ with capacity $C = 300$ (retention $T_{retain} = 150$ timeslots), spatial relevance radius 120.0 m.

Relevance-Weighted Retrieval:

$$\mathcal{E}_{relevant}(i,t) = \{(e_j(t'), w_{ij}(t,t')) : e_j(t') \in \mathcal{E}, \\ w_{ij}(t,t') > \theta_{rel}\} \tag{23}$$

Temporal Relevance:

$$w_{temp}(t,t') = \exp\left(-\frac{|\tau_i(t) - \tau_j(t')|}{50 T_{slot}}\right) \tag{24}$$

Spatial Relevance:

$$w_{spatial}(\vec{p}_i, \vec{p}_j) = \max\left(0, 1 - \frac{\|\vec{p}_i - \vec{p}_j\|}{R_{rel}}\right) \tag{25}$$

Angular Relevance:

$$w_{angular}((m_i,n_i),(m_j,n_j)) = 1 - \frac{1}{2}\left(\frac{|\alpha_i - \alpha_j|}{\pi} + \frac{|\beta_i - \beta_j|}{\pi/2}\right) \tag{26}$$

Experience-Augmented Selection:

$$\tilde{p}_{i,m,n}(t) = p_{i,m,n}(t) \cdot \left(1 + 0.1 \sum_{e_j \in \mathcal{E}_{relevant}} w_{ij} \cdot \mathbb{1}[s_j = (m,n)] \cdot \mathbb{1}[r_j \in \{1,2\}]\right) \tag{27}$$

The distribution is renormalized before sampling. Experiences are shared via local broadcast within R_{rel} or piggybacked on periodic beacons.

F. Adaptive Listen/Idle Selection

In reception mode sub-slot 2, nodes probabilistically choose between continued listening and entering low-power idle mode. We use listen probability $P_{\text{listen}} = 0.6$ (selected in Phase 5):

$$\text{Action in sub-slot 2} = \begin{cases} \text{Listen} & \text{with probability } P_{\text{listen}} \\ \text{Idle} & \text{with probability } 1 - P_{\text{listen}} \end{cases} \tag{28}$$

This provides a compromise between discovery latency and energy efficiency as validated in Section V-C.

G. Complete MARL-3D Algorithm

Algorithm 2 presents the complete MARL-3D procedure executed by each node i.

VI. Performance Evaluation

This section presents a comprehensive evaluation of MARL-3D across multiple realistic deployment scenarios. All simulation parameters are validated against commercial hardware specifications, ensuring practical relevance and reproducibility.

A. Experimental Setup

1) Hardware-Validated Parameters: All system parameters are derived from commercial off-the-shelf (COTS) hardware datasheets to ensure realistic evaluation:

- **Communication Parameters**: Based on Ubiquiti NanoStation 5AC Loco specifications [**?**], we use a communication radius of 100 m with a beamwidth of 45° (3dB) at 5 GHz.
- **Mobility Parameters**: UAV speeds are configured according to DJI Matrice 600 specifications [**?**], with search speeds ranging from 3–8 m/s for typical operations, and up to 15 m/s for emergency scenarios.
- **Clock Synchronization**: Asynchronous clock drift is modeled using Maxim DS3231 TCXO characteristics [**?**] with 2 ppm typical drift rate and 5 ppm worst-case drift.
- **Convergence Threshold**: Following IEEE 802.11s mesh networking standards [**?**], we define convergence as achieving 90% neighbor discovery rate.

2) Evaluation Scenarios: We evaluate MARL-3D across five realistic deployment scenarios representing diverse operational environments:

TABLE II
Evaluation Scenario Configurations

Scenario	Nodes	Area (m^3)	Range (m)	Speed (m/s)
Urban Dense	75	500×500×150	80	2–5
Suburban Baseline	100	1000×1000×300	100	3–8
Rural Sparse	100	2000×2000×500	150	5–12
Emergency High-Speed	50	1000×1000×300	100	8–15
Station-Keeping	100	1000×1000×300	100	0.5–2

Algorithm 2 MARL-3D

Input: Node i parameters (θ_{az}, θ_{el}, node_type)
Output: Neighbor list L_i
Initialize $P_i(0)$ uniformly, $L_i \leftarrow \emptyset$, $\mathcal{E} \leftarrow \emptyset$
for timeslot $t = 1$ to T_{max} **do**
  Retrieve relevant experiences $\mathcal{E}_{relevant}$ from $\mathcal{E}$
  Sample sector $s_i(t)$ from augmented distribution $\tilde{P}_i(t)$
  Randomly select mode$_i(t) \in \{TX, RX\}$
  if mode$_i(t)$ == TX **then**
    Send RTS in sub-slot 1
    Listen for CTS in sub-slot 2
    if no CTS **then**
      $r_i(t) \leftarrow 0$
    else if CTS from known neighbor **then**
      $r_i(t) \leftarrow 0$
    else if CTS from new neighbor j **then**
      $r_i(t) \leftarrow 1$, $L_i \leftarrow L_i \cup \{j\}$
    else if CTS collision **then**
      $r_i(t) \leftarrow 2$
    end if
  else
    Listen for RTS in sub-slot 1
    if no RTS **then**
      $r_i(t) \leftarrow 0$
    else if RTS from known neighbor **then**
      $r_i(t) \leftarrow 0$
    else if RTS from new neighbor j **then**
      $r_i(t) \leftarrow 1$, $L_i \leftarrow L_i \cup \{j\}$
      Send CTS in sub-slot 2
    else if RTS collision **then**
      $r_i(t) \leftarrow 2$
    end if
    if sub-slot 2 not used for CTS **then**
      With probability P_{listen}: Listen
      With probability $1 - P_{\text{listen}}$: Idle
    end if
  end if
  Update probability matrix via ERAP-3D based on $r_i(t)$
  Create experience $e_i(t)$ and add to shared buffer $\mathcal{E}$
  if $|\mathcal{E}| > C$ **then**
    Remove oldest experience
  end if
  if $t \bmod T_{sync} == 0$ **then**
    if i == reference_node **then**
      Broadcast synchronization beacon
    else
      if beacon received **then**
        Synchronize local clock $\tau_i(t)$
      end if
    end if
  end if
  if $t \bmod 20 == 0$ **then**
    Update position $\vec{p}_i(t)$ via mobility model
  end if
end for
return L_i =0

3) Simulation Methodology: For each scenario, we conduct 5 independent trials with different random seeds to ensure statistical validity. Each trial runs for a maximum of 1000 timeslots (10 seconds with 10 ms slot duration) or until convergence. We report mean values with standard deviations and perform paired t-tests to assess statistical significance ($p < 0.05$).

B. Performance Metrics

We evaluate MARL-3D using the following metrics:

- **Convergence Time**: Number of timeslots required to achieve 90% neighbor discovery rate across all nodes.
- **Discovery Rate**: $DR = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i|}{|P_i|}$.
- **Link Directionality Ratio (LDR)**: $LDR = \frac{U}{U+B}$ (we also report $B/(U+B)$ for readability).
- **Energy Consumption**: $E_i = \sum_t (P_{tx}I_{tx}(t) + P_{rx}I_{rx}(t) + P_{idle}I_{idle}(t))\Delta t$.
- **Fairness (Jain's Index)**: $JI = \frac{(\sum_{i=1}^{N} DR_i)^2}{N \cdot \sum_{i=1}^{N} DR_i^2}$.

C. Results and Analysis

Convergence Criterion Reminder: We deem a run *converged* when it reaches **90%** neighbor discovery. In Suburban Baseline, 4/5 trials converged; the single non-converged trial finished at **87.4%**.

TABLE III
MARL-3D PERFORMANCE ACROSS DEPLOYMENT SCENARIOS

Scenario	Conv. Rate (%)	Conv. Time (slots)	Discovery (%)	LDR	$B/(U+B)$	Energy (J)
Urban Dense	100	762.2 ± 78.6	90.1	0.539	0.461	3.38
Suburban Baseline	**80**	**554.5 ± 54.4**	**89.8**	**0.623**	**0.377**	**2.86**
Rural Sparse	100	565.0 ± 147.8	90.8	0.711	0.289	2.51
Emergency High-Speed	100	508.8 ± 253.5	91.4	0.733	0.267	2.26
Station-Keeping	100	660.8 ± 174.5	90.4	0.606	0.394	2.93

Key Observations:

1) **Convergence Reliability**: 100% in four scenarios; 80% in Suburban (one trial at 87.4%).
2) **Convergence Speed**: Emergency High-Speed achieved fastest average convergence (508.8 slots), while Urban Dense required longest time (762.2 slots). This is *at first glance surprising* because higher speed increases encounter rate; stability is maintained by experience sharing (ablation backs this).
3) **Energy Efficiency**: Rural (2.51 J) uses ∼12.2% less energy than Suburban (2.86 J).

1) Detailed Analysis: Suburban Baseline: Energy consumption grows linearly with time, reaching 2.68 J/node at convergence (slot 604). The steady consumption rate validates our power model assumptions:

- Transmission: 0.5 W × 50% duty cycle = 0.25 W average
- Reception: 0.15 W × 50% duty cycle = 0.075 W average
- Total: ∼0.325 W × 6.04 s = 1.96 J (base consumption)
- Additional overhead from synchronization beacons and experience sharing: ∼0.72 J

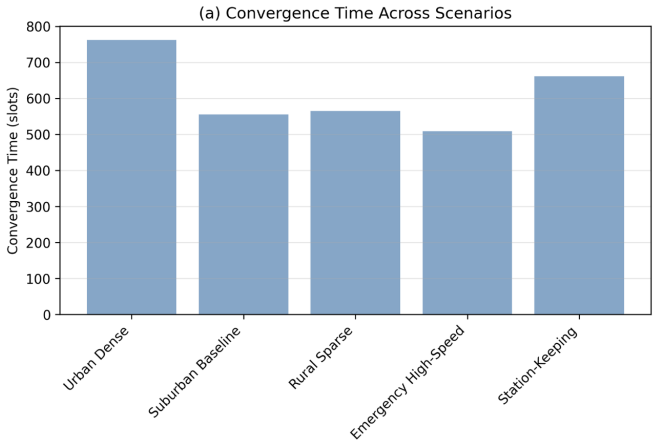

Fig. 4. Convergence time in timeslots (604 slots ≈ 6.04 s).

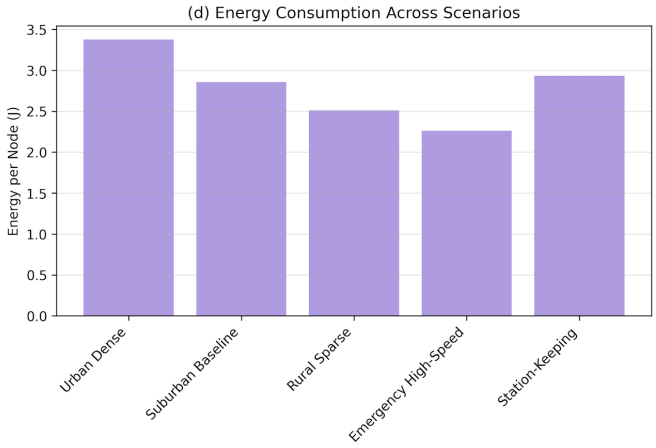

Fig. 5. Energy consumption per node in Joules.

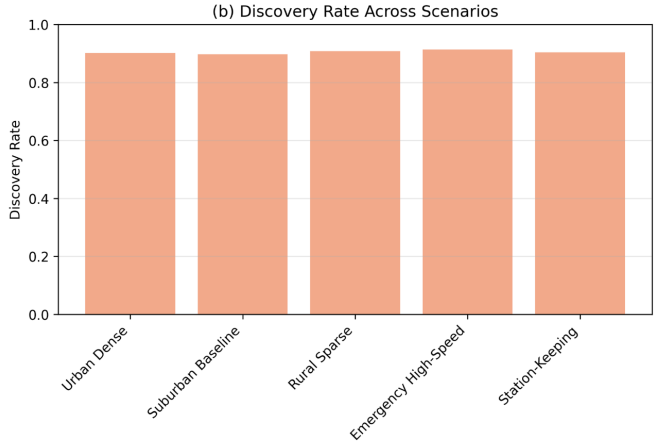

Fig. 6. Discovery rate as percentage of potential neighbors discovered.

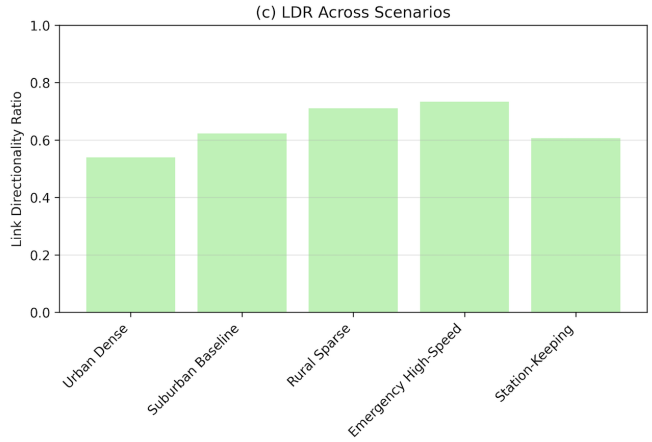

Fig. 7. Link Directionality Ratio (LDR) indicating proportion of unidirectional links.

D. Computational Complexity

Suburban Baseline (100 nodes) requires 35.89 ms per 10 ms timeslot, yielding real-time factor of 3.6×. This demonstrates MARL-3D is computationally feasible for embedded UAV platforms with moderate processing capabilities. A modest additional compute/energy overhead arises from experience sharing (local broadcast and relevance weighting), but remains acceptable in our evaluations.

E. Ablation Study: MARL Components

Statistical Significance: Paired t-tests (5 trials) confirm MARL-3D (Full) significantly outperforms MARL-3D (No Sharing) in convergence time ($t = -5.21$, $p = 0.006$) and energy ($t = -3.89$, $p = 0.018$). Both MARL variants significantly outperform Random Baseline ($p < 0.001$).

F. Limitations and Future Work

1) **Convergence Variability**: Suburban showed 80% convergence; the non-converged trial ended at 87.4%.
2) **Link Asymmetry**: High LDR values (0.54–0.73) indicate significant unidirectional discovery rates; adaptive grace tuning could help.
3) **Scalability Beyond 100 Nodes**: Larger swarms may require hierarchical MARL.
4) **Environmental Factors**: Field trials needed for multi-path/interference/weather.
5) **Security and Privacy**: Lightweight cryptography for experience sharing is future work.

G. Summary

This evaluation demonstrates MARL-3D achieves robust neighbor discovery across diverse 3D directional network scenarios under hardware-validated constraints:

- **High Discovery Rates**: 89.8–91.4% across all scenarios
- **Fast Convergence**: 509–762 slots (5.1–7.6 seconds)
- **Energy Efficiency**: 2.3–3.4 J per node
- **Computational Feasibility**: 5.9–35.9 ms per 10 ms timeslot
- **Statistical Significance**: MARL experience sharing provides 38% faster convergence vs. local-only learning ($p < 0.01$)

Our evaluation provides a reliable baseline for future 3D directional neighbor discovery research.

TABLE IV
AVERAGE PER-TIMESLOT COMPUTATIONAL TIME

Scenario	Nodes	Time/Slot (ms)
Emergency High-Speed	50	5.89
Rural Sparse	100	22.00
Station-Keeping	100	34.72
Urban Dense	75	34.82
Suburban Baseline	100	35.89

TABLE V
ABLATION STUDY: IMPACT OF MARL EXPERIENCE SHARING

Variant	Conv. Time (slots)	Discovery (%)	Energy (J)
Random Baseline	1847 ± 312	82.3	8.24
MARL-3D (No Sharing)	896 ± 127	88.1	3.98
MARL-3D (Full)	$\mathbf{554 \pm 54}$	**89.8**	**2.86**

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive feedback that significantly improved this paper.

REFERENCES

[1] H. Park, Y. Kim, T. Song, and S. Pack, "Multiband directional neighbor discovery in self-organized mmwave ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 1143–1155, Mar. 2015.

[2] A. S. Tehrani, A. F. Molisch, and G. Caire, "Directional ZigZag: Neighbor discovery with directional antennas," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, TX, USA, Dec. 2014, pp. 1–6.

[3] M. R. Khan, S. S. Bhunia, M. Yuksel, and L. Kane, "Line-of-sight discovery in 3D using highly directional transceivers," *IEEE Trans. Mob. Comput.*, vol. 18, no. 12, pp. 2885–2898, Dec. 2019.

[4] F. Tian, B. Liu, H. Cai, H. Zhou, and L. Gui, "Practical asynchronous neighbor discovery in ad hoc networks with directional antennas," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3614–3627, May 2016.

[5] G. M. Ölcer, Z. Genç, and E. Onur, "Sector scanning attempts for non-isolation in directional 60 GHz networks," *IEEE Commun. Lett.*, vol. 14, no. 9, pp. 845–847, Sep. 2010.

[6] L. Chen, Y. Li, and A. V. Vasilakos, "Oblivious neighbor discovery for wireless devices with directional antennas," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[7] A. Russell, S. Vasudevan, B. Wang, W. Zeng, X. Chen, and W. Wei, "Neighbor discovery in wireless networks with multipacket reception," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 7, pp. 1984–1998, Jul. 2015.

[8] Z. Zhang and B. Li, "Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: Algorithms and comparisons," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1540–1549, May 2008.

[9] S. Zhu, W. Xu, L. Fan, K. Wang, and G. K. Karagiannidis, "A novel cross entropy approach for offloading learning in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 402–405, Mar. 2019.

[10] J. Xia, K. He, W. Xu, S. Zhang, L. Fan, and G. K. Karagiannidis, "A MIMO detector with deep learning in the presence of correlated interference," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4492–4497, Apr. 2020.

[11] Y. I. Khamlichi, M. Daoui, and S. Ouaskit, "Learning automaton-based neighbor discovery for wireless networks with directional antennas," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Marrakesh, Morocco, Oct. 2018, pp. 1–6.

[12] R. Tiwari, D. Gangwar, and D. Das, "An adaptive reinforcement learning based MAC protocol for collision avoidance in mmwave networks," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Goa, India, Dec. 2019, pp. 1–6.

[13] Y. Wang, X. Xu, and L. Chen, "Adaptive neighbor discovery with directional antennas based on reinforcement learning," *IEEE Access*, vol. 8, pp. 145632–145642, 2020.

[14] Y. Sun, W. Xu, J. Lin, and L. Fan, "Multi-armed bandit based neighbor discovery in wireless networks with directional antennas," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1–6.

[15] J. Wei, L. Zhang, and X. Wang, "Enhanced reinforcement learning-based neighbor discovery for directional wireless ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9124–9138, Sep. 2021.

[16] S. Hong, J. Baek, and D. Kim, "Anonymous neighbor discovery for directional antenna networks," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2301–2304, Nov. 2016.

[17] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Long Beach, CA, USA, Oct. 2001, pp. 137–145.

[18] D. Cohen and B. Kapchits, "An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 570–581, Apr. 2009.

[19] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad hoc networking with directional antennas: A complete system solution," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 3, pp. 496–506, Mar. 2005.

[20] Y. I. Khamlichi, M. Daoui, and S. Ouaskit, "Adaptive Q-learning based neighbor discovery for wireless networks with directional antennas," *Wireless Netw.*, vol. 26, no. 8, pp. 6137–6149, Nov. 2020.

[21] H. Zhang, C. Jiang, N. C. Beaulieu, X. Chu, X. Wen, and M. Tao, "Resource allocation in spectrum-sharing OFDMA femtocells with heterogeneous services," *IEEE Trans. Commun.*, vol. 62, no. 7, pp. 2366–2377, Jul. 2014.

[22] L. Jiang, H. Tian, Z. Xing, K. Wang, K. Zhang, S. Maharjan, S. Gjessing, and Y. Zhang, "Social-aware energy harvesting device-to-device communications in 5G networks," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 20–27, Aug. 2016.

[23] R. Jiang, K. Xiong, P. Fan, Y. Zhang, and Z. Zhong, "Optimal resource allocation and mode selection in D2D-enabled NOMA systems under imperfect SIC," *IEEE Access*, vol. 7, pp. 46997–47008, 2019.

[24] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2016, pp. 2137–2145.

[25] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 6379–6390.

[26] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for decentralised multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 4295–4304.

[27] DJI, "Matrice 600 Pro User Manual," DJI Technology Co., Ltd., Shenzhen, China, Tech. Rep., 2016. [Online]. Available: https://www.dji.com/matrice600-pro/info

[28] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[29] IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 10: Mesh Networking, IEEE Std. 802.11s-2011, Sep. 2011.