# Cooperative Multi-LiDAR Deployment using Decentralized Multi-Agent Reinforcement Learning

Hyeong Jun Park
*Department of Artificial Intelligence*
*Hanyang University*
Seoul, Republic of Korea
phj990608@hanyang.ac.kr

Chang Mook Kang
*Department of Electrical Engineering*
*Hanyang University*
Seoul, Republic of Korea
kcm0728@hanyang.ac.kr

***Abstract*—Optimal placement of multiple LiDAR sensors is critical for autonomous driving, robotics, and smart city applications. This study compares three multi-agent coordination strategies—a local-search Greedy baseline, Independent Q-Learning (IQL), and Deep Q-Network (DQN)—for maximizing coverage in discrete, obstacle-rich environments. Experiments with 1–5 LiDAR sensors across varying obstacle densities show that IQL achieves the highest average coverage (89.5%) with low variance, while DQN exhibits irregular scaling and high sensitivity to environment changes. Our reward shaping strategy targets 30-40% sensor overlap to support inter-sensor coordination requirements. We additionally include a zero-shot evaluation for DQN as a transfer probe; the results suggest that generalization remains challenging in this discrete coordination setting. Results indicate that computationally efficient algorithms such as IQL can outperform complex deep learning approaches in robustness, scalability, and computational efficiency. These findings provide practical guidelines for designing multi-sensor systems in dynamic environments such as autonomous vehicles and urban infrastructure.**

## I. Introduction

Optimal placement of multiple Light Detection and Ranging (LiDAR) sensors is fundamental to numerous applications, including autonomous vehicles, smart city infrastructure, and robotic systems [1], [2]. The challenge lies in maximizing environmental coverage while managing sensor overlap and ensuring robust performance in complex, obstacle-rich environments.

Traditional approaches to sensor placement often rely on heuristic methods or evolutionary algorithms [3]. However, the emergence of multi-agent reinforcement learning (MARL) offers promising alternatives for addressing the inherent complexity of coordinated sensor deployment. We investigate how different MARL paradigms perform for multi-LiDAR placement optimization, and additionally report a zero-shot DQN setting as a transfer experiment to probe behavior across environmental configurations.

In certain multi-sensor applications, controlled overlap between adjacent sensors provides operational benefits. For instance, multi-sensor tracking systems require overlap regions for reliable object hand-off, while safety-critical deployments may prioritize redundancy. Excessive overlap, however, leads to inefficient resource utilization. Accordingly, this work incorporates a reward shaping strategy that explicitly targets a 30–40% overlap range to balance coordination potential with coverage efficiency.

We present a systematic comparison of three representative approaches: a greedy local-search method as a deterministic baseline, Independent Q-Learning (IQL) for decentralized coordination, and a Deep Q-Network (DQN) for handling higher-dimensional state representations with zero-shot transfer evaluation.

The experimental results reveal pronounced performance differences and provide insight into when and why particular coordination strategies succeed or fail. As the number of sensors increases, the placement problem rapidly becomes a high-dimensional coordination task, where balancing coverage and overlap requires increasingly sophisticated multi-agent strategies. While MARL offers a principled framework for learning such coordination through interaction, its effectiveness varies substantially across algorithmic paradigms and evaluation settings.

This work presents a comprehensive evaluation framework for multi-agent LiDAR placement in discrete, obstacle-rich environments, enabling systematic comparison across coordination strategies and environmental complexities. Through extensive experiments, we provide a detailed empirical analysis of decentralized tabular learning and deep reinforcement learning approaches under identical deployment constraints, together with a unified evaluation protocol that includes a supplementary zero-shot DQN analysis to probe robustness under distribution shift. In addition, we introduce an overlap-aware reward shaping strategy that explicitly treats controlled redundancy (30–40%) as a tunable design variable for coordination-aware sensor deployments. Taken together, these results provide practical insights into the design of robust and scalable multi-sensor systems for robotics and autonomous infrastructure.

While IQL and DQN themselves are standard multi-agent reinforcement learning techniques, the novelty of this work lies in the explicit formulation of overlap as a band-targeted coordination objective, a discrete visibility-based LiDAR placement formulation tailored to obstacle-rich environments, and a unified evaluation protocol; as part of this protocol, we additionally report a DQN zero-shot transfer experiment that highlights robustness and generalization trade-offs in multi-agent settings.

## II. Related Work

### A. Multi-Agent Reinforcement Learning

Multi-agent reinforcement learning has gained significant attention for coordinating multiple autonomous agents [4]. Independent Q-Learning, despite its theoretical limitations regarding convergence guarantees in multi-agent settings, has shown practical success in various domains due to its scalability and simplicity [5]. Recent work has explored the effectiveness of independent learning in multi-agent coordination tasks, often finding that simple approaches can outperform more complex centralized methods [6]. In addition, deep RL baselines such as DQN [9] and value decomposition methods VDN/QMIX [10], [11] are widely used for cooperative control.

The challenge in multi-agent reinforcement learning lies in the non-stationary environment each agent faces due to other agents' simultaneous learning. This non-stationarity can lead to convergence issues and suboptimal solutions. However, in practice, independent learning approaches often demonstrate robust performance when combined with appropriate reward shaping and environmental structure.

### B. Zero-Shot Transfer Learning

Zero-shot learning in reinforcement learning focuses on transferring learned policies to unseen environments without additional training [8]. This approach is particularly relevant for sensor placement optimization where deployment environments may differ significantly from training conditions. The effectiveness of zero-shot transfer depends on the similarity between training and testing environments, as well as the generalization capabilities of the learned representations. Approaches such as Other-Play aim to induce conventions that transfer across partners and seeds [15].

Recent advances in deep reinforcement learning have shown promise for transfer learning across different domains, but the effectiveness in multi-agent discrete coordination tasks remains underexplored. The discrete nature of sensor placement problems presents unique challenges for neural network generalization compared to continuous control tasks.

### C. Sensor Placement Optimization

Classical sensor placement approaches include integer programming formulations [7] and evolutionary algorithms [3]. These methods typically require known environment models and may not scale well to dynamic or complex environments. Recent advances have incorporated deep learning methods, though their effectiveness varies significantly across problem domains and environmental conditions [8].

LiDAR coverage optimization has been addressed in various contexts, from autonomous driving applications [2] to smart city monitoring systems [1]. Ye et al. [3] proposed methods for roadside LiDAR placement that account for blind spots and overlapping coverage regions, noting that strategic overlap can reduce detection dead zones. However, these works primarily focus on maximizing coverage without explicitly controlling overlap ratios as a coordination parameter.

Comprehensive comparisons of reinforcement learning approaches for multi-LiDAR coordination remain limited, particularly regarding the trade-offs between different algorithmic approaches, zero-shot transfer capabilities, and their robustness across varying environmental conditions.

## III. Methodology

### A. Problem Formulation

We consider an $M \times N$ discrete grid with obstacle cells $\mathcal{O}_{\text{obs}}$ and free cells $\mathcal{F}$. Each LiDAR pose is represented by $(p, d)$, where $p \in \mathcal{F}$ is a cell location and $d \in \mathcal{D}$ is a discretized orientation.

*a) Deterministic visibility-based coverage:* We use a binary, deterministic visibility model. A free cell $c \in \mathcal{F}$ is covered by a pose $(p, d)$ if it satisfies range, field-of-view (FOV), and line-of-sight (LOS):

$$\mathcal{C}(p,d) = \{c \in \mathcal{F} \mid \text{inRange}(c;p) \wedge \text{inFOV}(c;p,d) \wedge \text{LOS}(c;p)\}. \tag{1}$$

Here, $\text{inRange}(c;p)$ denotes $\|c - p\| \leq r_{\max}$ and $\text{inFOV}(c;p,d)$ denotes that the bearing of $c$ from $p$ lies within the LiDAR FOV centered at $d$. LOS is computed by discrete ray-casting (DDA): a ray from $p$ is traversed cell-by-cell and terminates at the first obstacle (or at the maximum range). Cells behind the first obstacle are occluded.

*b) Objective:* The goal is to select $K$ LiDAR poses to maximize the fraction of covered free cells:

$$\max_{\{(p_i,d_i)\}_{i=1}^{K} \subseteq \mathcal{F} \times \mathcal{D}} \frac{\left| \bigcup_{i=1}^{K} \mathcal{C}(p_i, d_i) \right|}{|\mathcal{F}|}. \tag{2}$$

where $\{p_i\}_{i=1}^{K} \subseteq \mathcal{F}$ are sensor locations, $\{d_i\}_{i=1}^{K} \subseteq \mathcal{D}$ are their orientations, and $\mathcal{C}(p_i, d_i) \subseteq \mathcal{F}$ denotes the set of free cells visible from the $i$-th LiDAR under the deterministic model in Eq. 1.

### B. Multi-Agent Algorithms

*1) Independent Q-Learning (IQL):* We adopt Independent Q-Learning (IQL) as a decentralized baseline, where each LiDAR sensor $i \in \{1, \ldots, K\}$ maintains and updates its own tabular action-value function $Q_i(s, a)$ to maximize a shared team objective.

**State and Action Space:** The state $s_i = (r, c, d)$ represents the discrete sensor pose, comprising the grid coordinates $(r, c)$ and a discretized orientation index $d \in \{1, \ldots, |\mathcal{D}|\}$. The action space $\mathcal{A}_i$ is discrete and low-dimensional (11 actions total): translation to one of the 8 neighboring cells (Moore neighborhood), rotation by one step (clockwise/counter-clockwise), or remaining stationary.

**Learning Mechanism:** The resulting per-agent state–action space has size $|\mathcal{S}| \times |\mathcal{A}| = (M \cdot N \cdot |\mathcal{D}|) \times 11$ ($60 \times 60 \times 24 \times 11$ in our setup), which makes tabular learning feasible. This avoids neural function approximation, which can be sensitive to representation and hyperparameter choices in discrete grid environments. Q-values are updated using the standard temporal-difference rule:

$$Q_i(s_t, a_t) \leftarrow (1-\alpha)Q_i(s_t, a_t)+\alpha\left[r_t + \gamma \max_{a'} Q_i(s_{t+1}, a')\right]. \tag{3}$$

We fix $\alpha = 0.10$ and $\gamma = 0.95$ in all IQL experiments. Exploration follows an $\epsilon$-greedy policy with $\epsilon$ decayed exponentially from $\epsilon_0 = 0.40$ to $\epsilon_{\min} = 0.03$.

*a) Why Independent Q-Learning Works Well in This Setting:* Although Independent Q-Learning (IQL) is known to suffer from non-stationarity in general multi-agent environments, several structural properties of the proposed multi-LiDAR placement problem mitigate these effects in practice. First, the environment is fully discrete with a low-dimensional state and action space, enabling stable tabular learning without function approximation. Second, agents are only weakly coupled: each LiDAR independently controls its pose, and inter-agent interaction occurs solely through a shared global reward signal rather than through direct state transitions or joint actions. Consequently, policy updates of one agent do not directly alter the transition dynamics perceived by others.

Moreover, the overlap-aware band-target reward provides an explicit coordination signal that reduces ambiguity in multi-agent credit assignment. By softly encouraging overlap within a predefined range, agents are guided toward complementary sensing configurations without requiring centralized training, value decomposition, or inter-agent communication. Empirically, this design yields stable convergence and low variance across random seeds, as reflected in Sec. IV. These results suggest that, for structured discrete deployment tasks with limited inter-agent coupling, IQL serves as a robust and computationally efficient coordination baseline.

*2) Deep Q-Network (DQN): Map-Specific and Zero-Shot Settings:* We implement Deep Q-Network (DQN) agents using the MATLAB Reinforcement Learning Toolbox with optional GPU acceleration. Unless otherwise stated, training hyperparameters are fixed across all experiments.

We evaluate DQN under two distinct settings. In the *map-specific* setting, a DQN agent is trained from scratch on each evaluation map and tested on the same map after convergence, representing within-environment performance. In the *zero-shot* setting, DQN agents are pre-trained on a disjoint set of random maps and evaluated on unseen test maps without any further learning or fine-tuning; all zero-shot evaluations use frozen network parameters. For $K > 1$, we train one DQN policy per agent using the same action set and a shared team reward.

*3) Greedy (Local-Search) Baseline:* We use a *local-search greedy* baseline. Let the joint state be $s_t = \{(p_i^t, d_i^t)\}_{i=1}^K$. Define the coverage count map as $C_{\text{map}}(c; s) = \sum_{i=1}^K \mathbf{1}\{c \in \mathcal{C}(p_i, d_i)\}$, and the normalized coverage and overlap ratio as

$$\begin{aligned} C(s) &= \frac{\left|\bigcup_{i=1}^K \mathcal{C}(p_i, d_i) \cap \mathcal{F}\right|}{|\mathcal{F}|}, \\ O(s) &= \frac{\left|\{\, c \in \mathcal{F} \mid C_{\text{map}}(c; s) > 1 \,\}\right|}{\left|\{\, c \in \mathcal{F} \mid C_{\text{map}}(c; s) \geq 1 \,\}\right|}. \end{aligned} \tag{4}$$

Note that $O(s) \equiv r_{\text{ov}}(s)$ in Eq. 10. For sensor $i$ and action $a \in \mathcal{A}_i$ (move/rotate/stay), the one-step myopic score is

$$\begin{aligned} J_i(s, a) = & \left[C(\mathcal{T}_i(s, a)) - C(s)\right] \\ & - w_{\text{move}}\,\mathbf{1}_{\text{move}}(a) - w_{\text{rot}}\,\mathbf{1}_{\text{rot}}(a) \\ & - w_{\text{invalid}}\,\mathbf{1}_{\text{invalid}}(a) - w_{\text{collide}}\,\Xi(\mathcal{T}_i(s, a)) \\ & + R^{\text{band}}_{\text{overlap}}(O(\mathcal{T}_i(s, a))). \end{aligned} \tag{5}$$

where $\mathcal{T}_i(s, a)$ applies $a$ only to sensor $i$ (others fixed), $\mathbf{1}_{(\cdot)}$ are penalty indicators, $\Xi(\cdot)$ counts the total number of colliding sensor pairs (i.e., two sensors occupying the same grid cell) in the resulting joint state, and $R^{\text{band}}_{\text{overlap}}$ is the band-target overlap reward (Eq. 10). Penalty weights: $w_{\text{move}} = 0.01$, $w_{\text{rot}} = 0.003$, $w_{\text{invalid}} = 0.02$, $w_{\text{collide}} = 0.02$.

At each time step $t$, we perform coordinate-wise greedy updates over sensors:

$$\begin{aligned} a^*_{t,i} &\in \arg\max_{a \in \mathcal{A}_i} J_i\big(s_t^{(i-1)}, a\big), \\ s_t^{(i)} &= \mathcal{T}_i\big(s_t^{(i-1)}, a^*_{t,i}\big) \end{aligned} \tag{6}$$

with $s_t^{(0)} = s_t$ and $s_{t+1} = s_t^{(K)}$. We stop early if $C(s_{t+1}) \geq \tau$, where $\tau$ is set to the coverage achieved by the reference static greedy solution in Eq. 7. This procedure is purely local/myopic and does not re-optimize previously placed sensors; therefore we do *not* claim the set-cover style submodular guarantees that apply to classic static sequential greedy.

*a) Static Greedy (reference only):* For context, we also compute the classic static sequential greedy selection used only as a reference for setting the early-stop threshold $\tau$:

$$\begin{aligned} (p_{k+1}, d_{k+1}) = \arg & \max_{(p,d) \in \mathcal{F} \times \mathcal{D}} \\ & \left| \mathcal{C}(p, d) \setminus \bigcup_{i=1}^{k} \mathcal{C}(p_i, d_i) \right| \end{aligned} \tag{7}$$

It is not the baseline plotted in our figures.

## C. Experimental Framework

*1) Environment Setup and Experimental Design:* The experimental environment consists of a $60 \times 60$ grid, representing a 3.0m × 3.0m area with 5cm resolution. Each LiDAR sensor has a 70.4° field of view with 15° directional increments, providing 24 possible orientations. Three obstacle configurations (1, 2, and 4 obstacles) are evaluated to assess performance across varying environmental complexities.

The environment setup incorporates realistic constraints including line-of-sight occlusion, sensor range limitations, and discrete positioning. Line-of-sight is computed via grid-based ray casting (DDA), where rays are cast uniformly across each sensor's field of view and terminate upon encountering an obstacle or reaching the maximum range. While we use explicit grid-based ray casting (DDA) for LOS evaluation, recent work has explored occupancy grid mapping approaches that reduce or avoid explicit ray casting for efficiency [13]. We do not use such alternatives in this work. To reduce runtime, we precompute and cache the visibility set $\mathcal{C}(p, d)$ for each candidate pose $(p, d)$ and reuse it during training and evaluation.

*2) Zero-Shot Learning Framework:* To evaluate transfer learning capabilities and generalization robustness, we implement a comprehensive zero-shot experimental framework specifically for DQN agents:

**Training Phase:** DQN agents are pre-trained on 5 randomly generated training maps with obstacle sizes ranging from 5×5 to 12×12 cells. The training environments use random obstacle placement with controlled density to ensure diverse learning experiences.

**Testing Phase:** Pre-trained agents are evaluated without additional training on two distinct test scenarios: (1) fixed deterministic maps with 3 different obstacle configurations, and (2) random test maps with obstacle sizes ranging from 6×6 to 18×18 cells to assess generalization to different obstacle scale distributions.

**Evaluation Protocol:** The framework uses disjoint random seeds between training and testing phases to ensure no overlap between training and evaluation scenarios. Performance is measured using best coverage achieved on fixed maps and mean ± standard deviation across multiple random test scenarios. We follow common cooperative MARL evaluation practices (multiple seeds, fixed episode budgets) popularized by SMAC [12].

*3) Reward Structure with Overlap Control:* We define $R_{\text{coverage}}$ as the incremental increase in the normalized covered area between consecutive time steps. The reward function is designed to balance coverage maximization with controlled overlap:

$$R_{\text{total}} = R_{\text{coverage}} - \frac{1}{K}\sum_{i=1}^{K}\sum_{j\in\mathcal{P}} w_j\,\mathbf{1}_j(a_i) + R_{\text{overlap}}^{\text{band}}. \tag{8}$$

where $\mathcal{P} = \{\text{move, rot, invalid, collide}\}$ denotes penalty types and $\mathbf{1}_j(a_i)$ is the indicator that agent $i$ executed a penalty-triggering action of type $j$. All penalty terms are averaged over $K$ agents. We use $w_{\text{move}} = 0.01$, $w_{\text{rot}} = 0.003$, $w_{\text{invalid}} = 0.02$, and $w_{\text{collide}} = 0.02$ in all experiments.

$$R_{\text{coverage}} = C(s_{t+1}) - C(s_t), \tag{9}$$

To maintain overlap within the target range $[\theta_{\text{low}}, \theta_{\text{high}}] = [0.30, 0.40]$, we adopt a band-target reward:

$$R_{\text{overlap}}^{\text{band}} = \begin{cases} -0.05 \cdot \frac{\theta_{\text{low}} - r_{\text{ov}}}{\max(\theta_{\text{low}}, \varepsilon)} & \text{if } r_{\text{ov}} < \theta_{\text{low}} \\ +0.10 \cdot \left(1 - \frac{2|r_{\text{ov}} - 0.35|}{0.10}\right) & \text{if } \theta_{\text{low}} \le r_{\text{ov}} \le \theta_{\text{high}} \\ -0.10 \cdot \frac{r_{\text{ov}} - \theta_{\text{high}}}{1 - \theta_{\text{high}}} & \text{if } r_{\text{ov}} > \theta_{\text{high}} \end{cases} \tag{10}$$

where $r_{\text{ov}}(s) \equiv O(s)$ denotes the overlap ratio (multi-covered cells divided by covered cells). This design encourages overlap near the 35% midpoint while penalizing both insufficient overlap (limiting coordination potential) and excessive overlap (wasting resources). We set a small constant $\varepsilon = 10^{-6}$. We apply $R_{\text{overlap}}^{\text{band}}$ only when $K \ge 2$.

For completeness, we define an optional soft minimum-separation penalty to discourage sensors from being placed too close:

$$P_{\text{prox}}(s) = \frac{1}{\binom{K}{2}} \sum_{i<j} \max\Big(0, d_{\min} - \|p_i - p_j\|\Big), \tag{11}$$

with minimum separation $d_{\min} = 0.6\,\text{m}$ (12 cells at 5cm resolution). In all experiments, we set $w_{\text{prox}} = 0$, as collision/invalid-action penalties were sufficient in our discrete grid setting.

*4) Evaluation Protocols:* Train/Val/Test maps are disjoint for DQN pre-training. DQN (zero-shot) is frozen at test time.

IQL is trained independently on each evaluation map under the same environment-step budget and is evaluated after training on that map. Unless noted, per-episode step budgets and the number of random seeds are matched across methods. We use $\Delta$-coverage as $R_{\text{coverage}}$ in all learning runs.

## IV. Experimental Results

**Evaluation note (fairness and intent):** To keep the primary comparison symmetric, we evaluate Greedy, IQL, and DQN (map-specific) on a per-map basis. Greedy requires no training, while IQL and DQN (map-specific) are trained and evaluated on the same map under a fixed step budget. Separately, we include DQN(zero-shot)—pre-trained on disjoint random maps and evaluated without adaptation on unseen maps—as an additional transfer experiment. This transfer experiment is not intended to replace per-map training in the head-to-head comparison, but to illustrate how a learned policy behaves under distribution shift.

Results are reported as mean±std over $N_{\text{seed}} = 5$ seeds. For IQL on fixed maps, each $(\#\text{obs}, K)$ uses $E\times M\times T = 10 \times 5 \times 800 = 4.0 \times 10^4$ environment steps. DQN(zero-shot) is pre-trained on $N_{\text{train}}{=}5$ random maps per $(\#\text{obs}, K)$ with the same per-episode step budget and kept frozen during evaluation.

Figs. 1 and 2 demonstrate the effectiveness of IQL-based LiDAR placement optimization across different complexity levels. In the single-agent scenario (Fig. 1), the sensor positions itself to maximize visibility around the central obstacle, achieving 68.03% coverage. The multi-agent coordination case (Fig. 2) shows four sensors working together to achieve comprehensive coverage of 97.06%, with each sensor focusing on distinct areas to minimize overlap while maximizing total coverage.

Figure 3 presents a comprehensive performance comparison across all algorithms and environmental configurations. The results clearly demonstrate IQL's consistent superiority across different sensor counts and obstacle densities. Table I provides a breakdown of performance across different obstacle densities, revealing how environmental complexity affects algorithm performance and the relative advantages of each approach.

Table II presents detailed performance results averaged across all obstacle configurations in fixed environments. The results demonstrate clear performance hierarchies and scaling behaviors across different sensor counts.

TABLE I
PERFORMANCE ANALYSIS BY OBSTACLE DENSITY (% MEAN ± STD)

| Algorithm | 1 Obstacle | 2 Obstacles | 4 Obstacles | Overall |
|---|---|---|---|---|
| Greedy | 56.2 ± 27.0 | 72.6 ± 28.3 | 67.9 ± 29.8 | 65.6 ± 29.2 |
| **IQL** | **89.3** ± **12.1** | **92.2** ± **7.3** | **87.0** ± **12.2** | **89.5** ± **11.0** |
| DQN(map-specific) | 64.9 ± 17.0 | 65.6 ± 20.9 | 54.3 ± 12.7 | 61.6 ± 18.0 |
| DQN(zero-shot) | 55.3 ± 18.3 | 55.5 ± 21.5 | 62.8 ± 20.5 | 57.9 ± 20.5 |

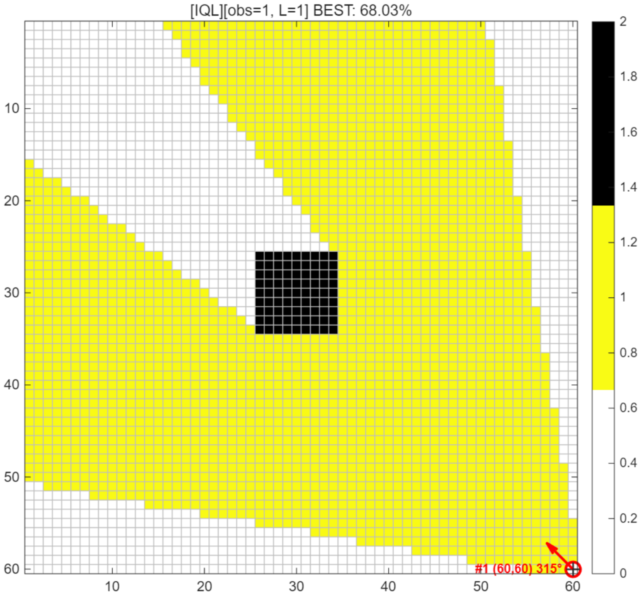

Fig. 1. Single-agent placement (1 obstacle, 1 LiDAR). Best coverage: 68.03%.

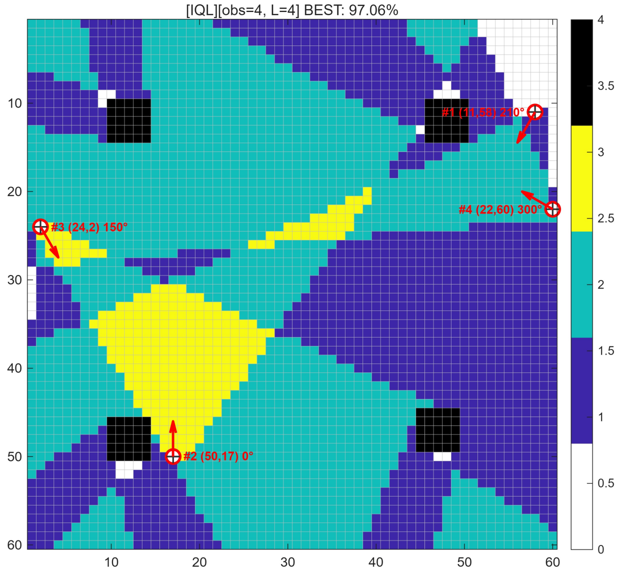

Fig. 2. Multi-agent coordination (4 obstacles, 4 LiDARs). Best coverage: 97.06%.

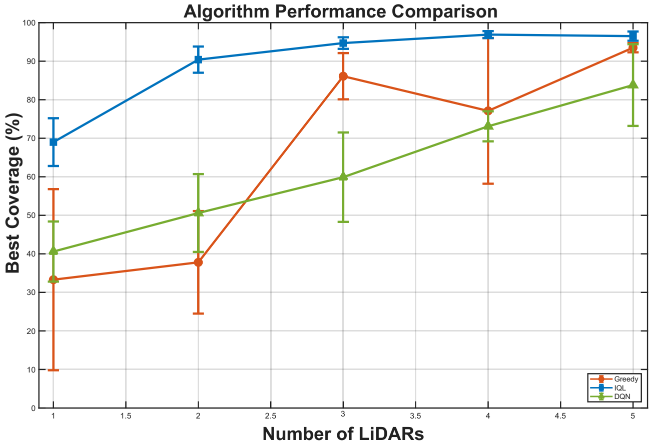

Fig. 3. Comprehensive performance comparison showing coverage performance across different numbers of LiDAR sensors and obstacle configurations. IQL (blue) consistently achieves the highest average coverage across the tested scenarios.

TABLE II
ALGORITHM PERFORMANCE COMPARISON ON FIXED ENVIRONMENTS (MEAN ± STANDARD DEVIATION %)

| LiDARs | Greedy | IQL | DQN(map-specific) |
|---|---|---|---|
| 1 | 33.3 ± 23.5 | **69.0** ± **6.2** | 40.6 ± 7.8 |
| 2 | 37.8 ± 13.3 | **90.4** ± **3.4** | 50.6 ± 10.1 |
| 3 | 86.1 ± 6.0 | **94.7** ± **1.5** | 59.9 ± 11.6 |
| 4 | 77.1 ± 18.9 | **96.9** ± **0.9** | 73.1 ± 3.9 |
| 5 | 93.5 ± 1.2 | **96.5** ± **1.2** | 83.8 ± 10.6 |
| **Average** | 65.6 ± 29.2 | **89.5** ± **11.0** | 61.6 ± 18.0 |

TABLE III
ADDITIONAL TRANSFER EXPERIMENT FOR DQN UNDER TWO SETTINGS: (I) DQN(MAP-SPECIFIC), TRAINED FROM SCRATCH ON EACH EVALUATION MAP, AND (II) DQN(ZERO-SHOT), PRE-TRAINED ON DISJOINT RANDOM MAPS AND EVALUATED WITHOUT FURTHER LEARNING ON TEST MAPS (DISTRIBUTION SHIFT) (%).

| LiDARs | IQL | DQN(map-specific) | DQN(zero-shot) |
|---|---|---|---|
| 1 | **69.0** ± **6.2** | 40.6 ± 7.8 | 33.1 ± 6.3 |
| 2 | **90.4** ± **3.4** | 50.6 ± 10.1 | 54.9 ± 15.5 |
| 3 | **94.7** ± **1.5** | 59.9 ± 11.6 | 67.3 ± 0.7 |
| 4 | **96.9** ± **0.9** | 73.1 ± 3.9 | 55.1 ± 21.8 |
| 5 | **96.5** ± **1.2** | 83.8± 10.6 | 78.9 ± 13.3 |
| **Average** | **89.5** ± **11.0** | 61.6 ± 18.0 | 57.9 ± 20.5 |

Table III reports the supplementary DQN zero-shot transfer results alongside per-map-trained baselines, illustrating that performance can be less consistent under distribution shift in this discrete coordination setting.

## V. Discussion

The superior performance of IQL demonstrates the remarkable effectiveness of decentralized learning approaches for multi-agent coordination tasks. By allowing each sensor to learn independently while sharing environmental rewards, IQL achieves effective coordination without the computational complexity and communication requirements of centralized or joint action methods. This approach proves particularly valuable in sensor network applications where communication bandwidth is limited and robustness to individual agent failures is critical.

As an additional transfer experiment, we evaluated DQN under a zero-shot setting to probe behavior across unseen maps. While DQN can achieve reasonable performance when trained per map, its behavior under distribution shift is less consistent in this discrete coordination problem, suggesting that reliable cross-map deployment would likely require environment-specific training or additional adaptation mechanisms [8], [15].

Both environment-specific trained and zero-shot DQN variants demonstrate issues including training instability, poor sample efficiency compared to tabular methods, high sensitivity to hyperparameters, and difficulty in learning effective coordination strategies. The computational complexity analysis reveals practical considerations: Greedy requires $O(K|\mathcal{F}|)$ per iteration, where $|\mathcal{F}|$ denotes the number of free grid cells; IQL uses $O(K|S||A|)$ memory for Q-tables; DQN incurs NN training and replay overhead.

Across increasing obstacle densities (1, 2, 4 obstacles), IQL maintains high coverage and relatively low variance compared to both Greedy and DQN. Performance gaps widen with complexity, suggesting that decentralized coordination becomes increasingly valuable as spatial complexity increases.

The band-target reward strategy (Eq. 10) provides a tunable design parameter for applications requiring controlled overlap. By targeting the 30-40% range in our experiments, the framework allows system architects to adjust $[\theta_{\text{low}}, \theta_{\text{high}}]$ bounds based on specific requirements such as fault tolerance or multi-target tracking coordination.

The findings suggest IQL as a strong default for multi-LiDAR systems due to its performance, variance, compute footprint, and robustness. For cross-environment deployment without retraining, current deep RL approaches are insufficient; environment-specific tuning or adaptive online learning should be considered.

Although wall-clock runtimes depend on hardware and implementation details, tabular IQL incurs negligible inference overhead compared to DQN, which requires forward passes through a neural network at each decision step.

## VI. Conclusion

We presented a comprehensive evaluation of MARL approaches for LiDAR placement, and additionally report a DQN zero-shot transfer experiment. Independent Q-Learning achieves superior average coverage and stability relative to deterministic baselines and deep RL approaches. Our band-target reward framework targets 30-40% controlled overlap, providing practical flexibility for applications requiring inter-sensor coordination. We also report an additional zero-shot transfer experiment for DQN, which suggests that robust transfer across unseen maps remains challenging in this setting. Simple, well-designed algorithms like IQL—paired with appropriate reward shaping—offer strong performance, computational efficiency, and reliability for real-world deployment.

## Acknowledgment

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) under Grant No. RS-2020-II201373 (Artificial Intelligence Graduate School Program, Hanyang University), No. RS-2025-25443597 (Development and demonstration of intelligent technologies for implementing autonomous patient transfer and mobility assistance systems), and No. RS-2025-13202969 (Development of hyper-personalized personal mobility service design).

## References

[1] Y. Li, *et al.*, “Is your LiDAR placement optimized for 3D scene understanding?” in *Advances in Neural Information Processing Systems*, 2024.

[2] W. Jiang, *et al.*, “Optimizing the placement of roadside LiDARs for autonomous driving,” in *Proc. ICCV*, 2023.

[3] T.-H. Kim, *et al.*, “Placement Method of Multiple Lidars for Roadside Infrastructure in Urban Environments,” *Sensors*, vol. 23, no. 21, p. 8808, 2023.

[4] A. Tampuu, *et al.*, “Multiagent Cooperation and Competition with Deep Reinforcement Learning,” *PLOS ONE*, vol. 12, no. 4, e0172395, 2017.

[5] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” in *Proc. ICML*, 1993.

[6] G. Papoudakis, *et al.*, “Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[7] A. Krause, *et al.*, “Near-optimal sensor placements in Gaussian processes,” *J. Mach. Learn. Res.*, vol. 9, 2008.

[8] P. Henderson, *et al.*, “Deep reinforcement learning that matters,” in *Proc. AAAI*, 2018.

[9] V. Mnih, *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, 2015.

[10] P. Sunehag, *et al.*, “Value-Decomposition Networks for Cooperative Multi-Agent Learning,” in *Proc. AAMAS*, 2018, pp. 2085–2087.

[11] T. Rashid, *et al.*, “QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning,” in *Proc. ICML*, 2018, pp. 4292–4301.

[12] M. Samvelyan, *et al.*, “The StarCraft Multi-Agent Challenge,” in *Proc. AAMAS*, 2019, pp. 2186–2188.

[13] Y. Cai, *et al.*, “Occupancy Grid Mapping Without Ray-Casting for High-Resolution LiDAR Sensors,” *IEEE Trans. Robot.*, vol. 40, pp. 172–192, 2024.

[14] N. Buchbinder and M. Feldman, “Constrained submodular maximization via a nonsymmetric technique,” *Math. Oper. Res.*, vol. 44, no. 3, pp. 988–1005, 2019.

[15] H. Hu, *et al.*, “Other-Play for Zero-Shot Coordination,” in *Proc. ICML*, 2020.