SEEK-Multi: Collaborative Multi-Agent Semantic Reasoning for Object Goal Navigation in Inspection Tasks

Adele Chinda
Georgia State University

***Abstract*—This paper addresses the fundamental challenge of collaborative multi-agent object-goal navigation for autonomous inspections in complex, real-world environments. While single-agent approaches to object-goal navigation have demonstrated considerable promise in recent years, scaling these methods to larger environments necessitates the coordination of multiple robots to achieve efficient coverage, faster task completion, and robust operation under uncertainty. We introduce SEEK-Multi, a comprehensive framework that extends semantic-guided object inspection to multi-robot systems through distributed belief sharing, collaborative planning, coordinated task allocation, and adaptive communication protocols. SEEK-Multi enables multiple agents to share semantic understanding and inspection findings through a distributed Relational Semantic Network (RSN) and a shared Dynamic Scene Graph (DSG), maintaining consistency across the team while accommodating communication constraints. We propose novel algorithms for collaborative exploration that leverage semantic priors, belief fusion using consensus protocols with provable convergence guarantees, and conflict-free task allocation based on auction mechanisms. Our extensive simulation analyses across diverse environment configurations demonstrate that SEEK-Multi achieves significant speedup over single-agent approaches while maintaining high success rates, with near-linear scaling efficiency for up to four agents and graceful degradation under communication failures. We validate our approach through comprehensive simulations including ablation studies, sensitivity analyses, and comparisons with state-of-the-art multi-agent coordination methods, demonstrating its practicality for real-world multi-robot inspection scenarios in industrial, search-and-rescue, and domestic environments. Code is available at: https://arrdel.github.io/seek-multi/**

I. Introduction

Consider a team of autonomous robots tasked with searching for and inspecting target objects across a large industrial facility. While a single robot can methodically search each area, the task completion time scales linearly with the environment size, making single-agent solutions impractical for time-critical applications. Deploying multiple robots offers the potential for significant speedup, but realizing this potential requires sophisticated coordination to avoid redundant effort and conflicting actions. This multi-agent object-goal navigation problem is crucial for time-sensitive applications such as emergency response, security patrols, industrial inspection, and search-and-rescue operations [53, 6, 41].

The deployment of multi-robot systems for inspection tasks has gained significant attention in recent years, driven by advances in sensing, communication, and computation [44, 43]. Industries ranging from manufacturing to energy production increasingly rely on autonomous inspection to reduce costs, improve safety, and enable continuous monitoring [29]. However, the transition from single-robot to multi-robot inspection introduces fundamental challenges in coordination, communication, and decision-making that require novel algorithmic solutions.

Multi-agent coordination for object-goal navigation presents unique challenges beyond the single-agent case. First, agents must efficiently partition the search space to minimize overlap while ensuring complete coverage, a problem that becomes increasingly complex as the number of agents and environment size grow. Second, agents must share observations and update their beliefs about object locations in a consistent manner, even with limited communication bandwidth and intermittent connectivity. Third, the planning framework must account for the actions and intentions of all agents to avoid conflicts and maximize team efficiency, requiring coordination mechanisms that scale gracefully. Fourth, the system must be robust to communication failures, agent heterogeneity, and dynamic environmental changes. Fifth, the framework must balance the benefits of coordination against its computational and communication costs, enabling operation in resource-constrained scenarios.

Recent work on single-agent semantic-guided navigation [25, 12, 48, 11] has demonstrated the value of incorporating prior knowledge and semantic reasoning into object search. The SEEK framework [25] introduced the Relational Semantic Network (RSN) for encoding object-room relationships and showed significant improvements over geometric coverage approaches. By leveraging semantic priors about where objects are likely to be found, SEEK enables efficient, informed search that outperforms uninformed exploration strategies. However, extending these methods to multi-agent settings requires addressing the challenges of distributed belief maintenance, coordinated planning, and efficient communication while preserving the semantic reasoning capabilities that make single-agent approaches effective.

The multi-agent extension of semantic navigation raises several research questions that motivate our work: How can semantic beliefs be efficiently shared and fused across multiple agents with potentially different observations? How should task allocation incorporate semantic priors while respecting

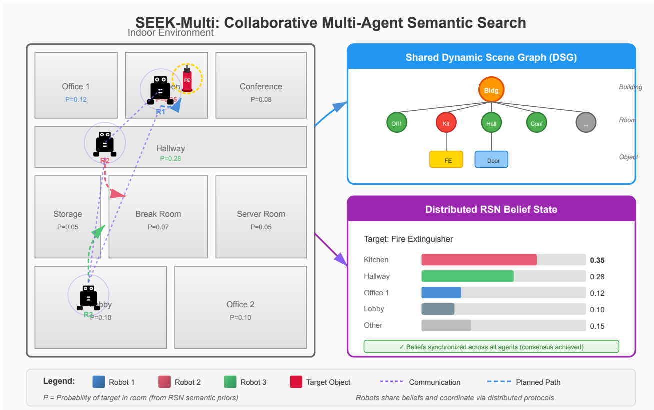

Fig. 1: SEEK-Multi enables multiple robots to collaboratively search for target objects by sharing semantic beliefs and coordinating their search strategies. Each agent maintains a local copy of the shared Dynamic Scene Graph (DSG) and Relational Semantic Network (RSN), with updates propagated through a distributed communication protocol. The framework supports both centralized and decentralized coordination modes.

coordination constraints? What communication protocols best balance information sharing with bandwidth limitations? How can the system maintain performance under communication failures and agent heterogeneity?

In this paper, we propose SEEK-Multi, a comprehensive framework for collaborative multi-agent object-goal navigation that extends the SEEK architecture to multi-robot teams. Our approach maintains distributed copies of the Dynamic Scene Graph (DSG) and Relational Semantic Network (RSN) across all agents, with efficient protocols for sharing updates and fusing beliefs. We introduce a collaborative planning algorithm that computes coordinated task assignments while accounting for agent positions, capabilities, and intentions. The framework supports both centralized and decentralized coordination modes, enabling deployment in various communication scenarios from reliable infrastructure networks to ad-hoc peer-to-peer connectivity.

The design of SEEK-Multi is guided by several key principles: **Semantic awareness**: All coordination mechanisms leverage semantic understanding to improve efficiency. **Scalability**: Algorithms and communication protocols scale gracefully with the number of agents. **Robustness**: The system degrades gracefully under communication failures and agent heterogeneity. **Flexibility**: The framework supports various coordination modes and can adapt to different deployment scenarios.

Our key contributions are:

1) We introduce SEEK-Multi, a comprehensive framework for collaborative multi-agent object-goal navigation using distributed semantic reasoning, supporting both centralized and decentralized coordination;
2) We propose a distributed belief fusion algorithm based on consensus protocols that enables agents to share and combine observations efficiently with provable convergence guarantees;
3) We design a collaborative planning algorithm that coordinates task allocation and path planning across multiple agents using auction-based mechanisms with semantic-aware bidding;
4) We develop an adaptive communication protocol that balances information sharing with bandwidth constraints and provides robustness to message loss;
5) We demonstrate through extensive simulation experiments that SEEK-Multi achieves near-linear speedup with multiple agents while maintaining high success rates across diverse environment configurations.

The remainder of this paper is organized as follows. Section II reviews related work in multi-robot coordination, semantic navigation, and distributed belief maintenance. Section III formally defines the multi-agent object-goal navigation problem. Section IV presents the SEEK-Multi architecture, including distributed semantic representations, belief fusion, and collaborative planning. Section V provides theoretical analysis of convergence and speedup properties. Section VI presents experimental results from simulation studies. Section VII discusses scalability, heterogeneity, and limitations. Section VIII concludes with directions for future work.

II. Related Works

Our work builds upon and integrates advances from several research areas: multi-robot coordination, multi-agent path planning, distributed belief maintenance, semantic navigation, and scene understanding. We review each area and position our contributions relative to existing work.

Multi-Robot Coordination: Multi-robot systems have been extensively studied for tasks including exploration [8, 60], coverage [14, 23], and search and rescue [37, 32]. Coordination strategies range from centralized approaches with a single decision-maker [24] to fully decentralized methods using local communication [38, 17]. Market-based approaches [20, 63] provide a middle ground, using auction mechanisms for task allocation while maintaining scalability. Recent work has explored learning-based coordination [21, 35], where agents learn coordination strategies through reinforcement learning.

The choice of coordination architecture significantly impacts system properties. Centralized approaches can achieve optimal coordination but require reliable communication to a central node and create a single point of failure [39]. Decentralized approaches offer robustness and scalability but may sacrifice optimality due to limited global information [61]. Hybrid approaches attempt to balance these tradeoffs by combining local decision-making with occasional global coordination [63]. Our work supports multiple coordination modes, allowing deployment in various scenarios.

Multi-Agent Exploration and Search: Coordinated exploration has been studied extensively, with frontier-based methods [60] forming a foundation for many approaches. Multi-robot extensions assign frontiers to robots based on distance, information gain, or other criteria [8, 50]. Recent work has incorporated semantic information into exploration [40], using object recognition to guide search toward promising areas.

Search tasks differ from exploration in that they seek specific targets rather than complete coverage. Multi-robot search has been studied for static targets [27], moving targets [16], and adversarial scenarios [58]. Probabilistic approaches maintain belief distributions over target locations and plan searches to maximize detection probability [7, 33]. Our work extends these concepts by incorporating semantic priors that capture object-room relationships.

Multi-Agent Path Planning: Coordinated path planning for multiple agents must balance efficiency with collision avoidance [51, 34]. Approaches include coupled planning that considers all agents jointly [49, 59], prioritized planning that plans sequentially [56, 9], and velocity-obstacle methods for dynamic environments [57, 4]. Conflict-based search (CBS) [49] has emerged as an efficient approach for optimal multi-agent path finding, using a two-level search that resolves conflicts lazily.

For continuous domains and longer time horizons, approaches based on potential fields [54], model predictive control [36], and learned policies [47] have shown promise. Our framework uses intention sharing and conflict resolution to enable efficient distributed planning without requiring coupled optimization over all agents.

Distributed Belief Maintenance: Maintaining consistent beliefs across multiple agents is fundamental to multi-robot perception [19, 3]. Consensus algorithms [38, 42] provide a principled approach to fusing estimates from multiple agents, with well-understood convergence properties. Distributed simultaneous localization and mapping (SLAM) [18, 15] addresses the related problem of building shared maps from distributed observations.

Belief fusion must account for correlations between agent observations to avoid overconfidence [31]. Covariance intersection [30] provides conservative fusion when correlations are unknown, while channel filters [13] track information flow to avoid double-counting. Our belief fusion approach uses weighted consensus with confidence tracking to balance these concerns in a computationally efficient manner.

Semantic Navigation and Scene Understanding: Recent advances in semantic navigation leverage foundation models for improved reasoning [12, 48, 62, 22]. These approaches use vision-language models to understand scene semantics and guide navigation toward likely target locations. The SEEK framework [25] demonstrated the value of encoding object-room relationships in a Relational Semantic Network, achieving significant improvements over geometric coverage approaches.

Scene graphs provide structured representations of environments that capture objects, rooms, and their relationships [45, 28, 2]. Dynamic Scene Graphs (DSG) [46] extend this to include temporal information and support real-time updates during robot operation. Multi-robot scene graph construction has been explored for collaborative mapping [10, 55], but integration with semantic search remains limited. Our work extends SEEK to multi-agent settings with distributed belief sharing, collaborative planning, and efficient communication protocols that maintain semantic scene graph consistency.

Communication in Multi-Robot Systems: Communication is fundamental to multi-robot coordination, with significant research on protocols, bandwidth management, and robustness [61, 41]. Approaches range from continuous communication assuming reliable infrastructure [8] to intermittent communication in bandwidth-limited scenarios [26]. Learning-based methods have explored communication protocol optimization [21, 52], allowing agents to learn what information to share.

Our communication protocol balances information sharing with bandwidth constraints, prioritizing high-value updates while maintaining robustness to message loss through redundancy and acknowledgment mechanisms.

III. Problem Formulation

We formally define the multi-agent object-goal navigation problem, including the environment model, agent capabilities, communication constraints, and performance metrics.

A. Environment Model

We consider a structured indoor environment represented as a topological-metric map. The environment is partitioned into a set of rooms $\mathcal{V} = \{v_1, \ldots, v_M\}$ connected by traversable edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each room v_j has associated attributes including:

- Semantic type $\ell(v_j) \in \mathcal{L}$ (e.g., office, kitchen, hallway)
- Geometric extent defining the room boundary
- Set of contained objects $\mathcal{O}(v_j)$
- Search time $\tau(v_j)$ required for thorough inspection

The environment graph $\mathcal{G}_{\text{env}} = (\mathcal{V}, \mathcal{E})$ may be partially or fully known a priori, depending on whether a floor plan is available. Edge weights $w(e)$ represent travel costs between adjacent rooms.

B. Multi-Agent Object-Goal Navigation

We consider a team of N robots $\mathcal{R} = \{r_1, \ldots, r_N\}$ operating in the environment. Each robot r_i has state $x_i \in X$, takes actions $u_i \in U$, and receives observations $z_i \in Z$. The state space $X = SE(2) \times \mathcal{V}$ includes the robot's pose and current room. The action space U includes navigation actions (move to adjacent room, move within room) and inspection actions (search for objects). The observation space Z includes object detections with associated confidence scores.

The robots communicate through a network with potentially limited bandwidth and range. We model the communication graph $\mathcal{G}^t_{\text{comm}} = (\mathcal{R}, \mathcal{E}^t_{\text{comm}})$ as potentially time-varying, where $(r_i, r_j) \in \mathcal{E}^t_{\text{comm}}$ if agents i and j can communicate at time t.

The objective is to locate a target object y_G of class y^l_G as quickly as possible. The target object is located in one of the rooms, with its location unknown to the robots. We assume the target is static and detectable with known sensor characteristics.

C. Observation Model

Each robot has a detection sensor with the following characteristics:

- Detection range d_{det}: maximum distance at which objects can be detected
- True positive rate p_{tp}: probability of detecting the target when in range
- False positive rate p_{fp}: probability of false detection per observation
- Position noise σ_p: uncertainty in detected object position

When robot r_i is in room v containing the target object and performs a search action, the observation model is:

$$P(z_i = \text{detect}|y_G \in v) = p_{\text{tp}} \cdot \mathbf{1}[\text{in detection range}] \tag{1}$$

For thorough search of a room (visiting all areas within detection range), the cumulative detection probability approaches p_d, the room-level detection probability.

D. Semantic Prior Model

We leverage semantic priors about object-room relationships, captured in a Relational Semantic Network (RSN). The RSN encodes the conditional probability of finding an object class in a room type:

$$P(y^l \in \ell) = \text{RSN}(y^l, \ell) \tag{2}$$

For example, $P(\text{fire extinguisher} \in \text{kitchen})$ captures the prior knowledge that fire extinguishers are commonly found in kitchens. These priors are learned from datasets of real indoor environments and can be updated based on domain-specific knowledge.

Given the room types in the environment, the prior probability of finding the target in each room is:

$$P_0(y_G \in v_j) = \frac{P(y_G^l \in \ell(v_j))}{\sum_{k=1}^{M} P(y_G^l \in \ell(v_k))} \tag{3}$$

E. Performance Metrics

We measure performance using the team's Success weighted by Path Length (SPL):

$$\text{SPL}_{\text{team}} = S \cdot \frac{l^*}{\max(\sum_{i=1}^{N} p_i, l^*)} \tag{4}$$

where $S \in \{0, 1\}$ indicates success, l^* is the optimal single-agent path length, and p_i is the path length traveled by robot r_i.

This metric captures both the success rate and the efficiency of the team's search. The denominator uses the sum of path lengths to account for the total resources consumed by the team. An alternative metric uses the maximum path length to focus on time-to-completion:

$$\text{SPL}_{\text{time}} = S \cdot \frac{l^*}{\max(\max_i p_i, l^*)} \tag{5}$$

We also measure:

- **Success rate (SR)**: Fraction of trials where target is found within time limit
- **Time to completion (TTC)**: Time steps until target is found
- **Coverage overlap**: Fraction of rooms searched by multiple agents
- **Communication efficiency**: Messages per successful search

F. Problem Statement

Problem 1 (Multi-Agent Object-Goal Navigation): Given a target object class y_G^l, initial robot states $\{x_1^0, \ldots, x_N^0\}$, environment graph $\mathcal{G}_{\text{env}}$, semantic priors from the RSN, and communication constraints, find a joint policy $\pi = (\pi_1, \ldots, \pi_N)$ that maximizes expected team SPL:

$$\begin{aligned} \pi^* = &\arg\max_{\pi} \mathbb{E}\left[\text{SPL}_{\text{team}}\right] \\ \text{s.t. } &x_i^{k+1} = f(x_i^k, \pi_i(b_i^k)), \quad \forall i \\ &b_i^{k+1} = \tau(b_i^k, \pi_i(b_i^k), z_i^{k+1}, \mathcal{M}_i^k) \\ &|\mathcal{M}_i^k| \leq B, \quad \forall i, k \end{aligned} \tag{6}$$

where b_i^k is robot i's belief state, $\mathcal{M}_i^k$ represents messages received from other robots, and B is the bandwidth constraint.

The belief state b_i^k includes:

- Probability distribution $P_i(y_G)$ over target location
- Local copy of the Dynamic Scene Graph $\mathcal{G}_i$
- Estimates of other agents' states and intentions
- History of searched rooms

G. Communication Model

Robots communicate through message passing with the following constraints:

- **Range**: Robot r_i can communicate with r_j if $\|x_i - x_j\| \leq d_{\text{comm}}$
- **Bandwidth**: Maximum B messages per time step per robot
- **Latency**: Messages arrive with delay δ
- **Reliability**: Messages are delivered with probability $1 - p_{\text{loss}}$

We define several message types with associated priorities:

- **Object detections** (priority 3): Location and confidence of detected objects
- **Belief updates** (priority 2): Probability distributions over rooms
- **Intentions** (priority 2): Planned actions for coordination
- **DSG updates** (priority 1): Changes to the shared scene graph
- **Heartbeats** (priority 0): Status and position updates

Higher priority messages are transmitted first when bandwidth is limited.

IV. SEEK-Multi Architecture

SEEK-Multi extends the single-agent SEEK architecture to multi-robot teams through three key components: distributed semantic representations, collaborative planning, and a communication protocol for belief sharing. Figure 2 illustrates the overall system architecture.

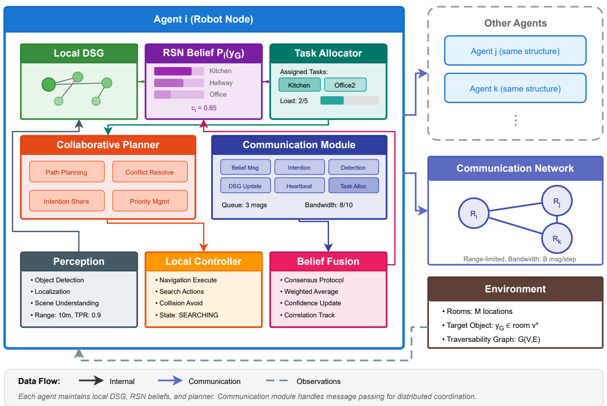

Fig. 2: SEEK-Multi system architecture. Each agent maintains local copies of the DSG and RSN beliefs, updated through observation and communication. The collaborative planner coordinates task allocation, while the communication module manages information sharing.

A. System Overview

Each robot in SEEK-Multi maintains the following components:

- **Local DSG**: A copy of the Dynamic Scene Graph representing the environment structure and discovered objects
- **Local RSN Belief**: Probability distribution over target location based on semantic reasoning
- **Task Allocator**: Component for coordinating room assignments with other agents
- **Local Planner**: Navigation planner for reaching assigned targets
- **Communication Module**: Interface for sending and receiving messages

The robots operate in a sense-plan-act loop, with the following phases each time step:

1) **Sense**: Receive observations from sensors and messages from other agents
2) **Update**: Update local DSG and beliefs based on new information
3) **Coordinate**: Exchange intentions and resolve conflicts
4) **Plan**: Compute actions to execute
5) **Act**: Execute planned actions
6) **Communicate**: Share updates with other agents

B. Distributed Semantic Representations

1) Shared Dynamic Scene Graph: The Dynamic Scene Graph (DSG) provides a hierarchical representation of the environment, capturing spatial relationships at multiple levels of abstraction [45]. In SEEK-Multi, each robot maintains a local copy of the DSG $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, with layers representing:

- **Object layer**: Individual objects with positions and semantic labels
- **Place layer**: Navigable locations within rooms
- **Room layer**: Semantic regions with room types
- **Building layer**: Overall structure and connectivity

The DSG is initialized from a common blueprint (floor plan) if available, or built incrementally through exploration. When robot r_i discovers new information (e.g., an object or updated traversability), it broadcasts an incremental update:

$$\Delta\mathcal{G}_i = \{(\text{op}, \text{node/edge}, \text{data}, t_i, c_i)\} \quad (7)$$

where op $\in \{\text{add}, \text{update}, \text{delete}\}$, t_i is a timestamp, and c_i is a confidence score.

Robots merge updates using timestamp-based conflict resolution with confidence weighting:

$$\mathcal{G}_i \leftarrow \text{Merge}(\mathcal{G}_i, \Delta\mathcal{G}_j) \quad \text{if } t_j > t_i^{\text{node}} \vee c_j > c_i^{\text{node}} \quad (8)$$

The merge operation handles several cases:

- **New nodes/edges**: Added directly to local DSG
- **Updated attributes**: Newer or higher-confidence values take precedence
- **Conflicting observations**: Resolved using confidence-weighted averaging for continuous attributes
- **Deletions**: Marked as deleted but retained for consistency checking

2) Distributed Relational Semantic Network: The Relational Semantic Network (RSN) predicts the probability of finding the target object in each room based on semantic priors. The RSN encodes relationships between object classes and room types learned from training data:

$$\text{RSN} : \mathcal{Y} \times \mathcal{L} \rightarrow [0, 1] \quad (9)$$

where $\mathcal{Y}$ is the set of object classes and $\mathcal{L}$ is the set of room types.

In SEEK-Multi, each robot maintains a local belief state over target location:

$$P_i(y_G) = \{P_i(y_G \in v_j) : v_j \in \mathcal{V}\} \quad (10)$$

The belief is initialized from the RSN prior:

$$P_i^0(y_G \in v_j) = \frac{\text{RSN}(y_G^l, \ell(v_j))}{\sum_k \text{RSN}(y_G^l, \ell(v_k))} \quad (11)$$

Beliefs are updated from local observations using Bayesian inference:

$$P_i(y_G \in v | z_i) = \frac{P(z_i | y_G \in v) \cdot P_i(y_G \in v)}{\sum_{v'} P(z_i | y_G \in v') \cdot P_i(y_G \in v')} \quad (12)$$

When robot r_i completes a thorough search of room v without finding the target:

$$P_i(y_G \in v | \neg\text{det}) = \frac{(1 - p_d) \cdot P_i(y_G \in v)}{1 - p_d \cdot P_i(y_G \in v)} \quad (13)$$

where p_d is the detection probability for a thorough search.

When a potential target is detected with confidence c_{det}:

$$P_i(y_G \in v | \text{det}) = \frac{c_{\text{det}} \cdot p_{\text{tp}} + (1 - c_{\text{det}}) \cdot p_{\text{fp}}}{Z} \quad (14)$$

where Z is the normalizing constant.

C. Distributed Belief Fusion

Robots share belief updates through a consensus-based fusion protocol that ensures beliefs converge across the team while accommodating communication delays and losses.

1) Weighted Consensus Fusion: When robot r_i receives belief $P_j(y_G)$ from robot r_j with confidence c_j, it updates its belief using weighted averaging:

$$P_i^{\text{new}}(y_G \in v) = \frac{c_i \cdot P_i(y_G \in v) + c_j \cdot P_j(y_G \in v)}{c_i + c_j} \quad (15)$$

The confidence is updated based on the number and quality of observations:

$$c_i^{\text{new}} = \min\left(c_{\max}, \sqrt{c_i^2 + c_j^2} + \alpha \cdot |\text{obs}_i^{\text{new}}|\right) \quad (16)$$

This update rule has several desirable properties:

- Beliefs from agents with more observations have higher influence
- Confidence grows sublinearly to prevent overconfidence
- The fusion is symmetric and associative

2) Periodic Synchronization: For environments with limited or intermittent communication, we use periodic synchronization:

$$P^{\text{sync}}(y_G \in v) = \frac{\sum_{j \in \mathcal{N}_i} c_j \cdot P_j(y_G \in v)}{\sum_{j \in \mathcal{N}_i} c_j} \quad (17)$$

where $\mathcal{N}_i$ is the set of robots in communication range of r_i.

Synchronization events are triggered:

- Periodically at fixed intervals
- When belief divergence exceeds threshold
- When agents come into communication range after disconnection

3) Handling Correlated Observations: When agents observe the same room or share information, their beliefs become correlated. To avoid overconfidence from double-counting, we track observation sources:

$$\text{Sources}_i(v) = \{(r_j, t_j) : r_j \text{ observed } v \text{ at time } t_j\} \quad (18)$$

When fusing beliefs, we discount contributions from already-incorporated observations:

$$w_j = c_j \cdot (1 - \text{Overlap}(\text{Sources}_i, \text{Sources}_j)) \quad (19)$$

D. Collaborative Planning

1) Joint MDP Formulation: We formulate the multi-agent planning problem as a factored Markov Decision Process (MDP) that decomposes across agents while capturing coordination requirements. The joint state $\mathcal{X} = (x_1, \ldots, x_N, s)$ includes robot states and the search state s (which rooms have been searched). The joint action $\mathcal{U} = (u_1, \ldots, u_N)$ specifies actions for all robots.

The transition probability factors as:

$$P(\mathcal{X}'|\mathcal{X}, \mathcal{U}) = P(s'|s, \mathcal{U}) \prod_{i=1}^{N} P(x_i'|x_i, u_i) \quad (20)$$

The reward function encourages finding the target while penalizing travel distance and coordination conflicts:

$$R(\mathcal{X}, \mathcal{U}) = R_{\text{find}} \cdot \mathbf{1}[\text{found}] - \sum_{i=1}^{N} c(u_i) - \lambda \cdot \text{Conflict}(\mathcal{U}) \quad (21)$$

where $\text{Conflict}(\mathcal{U})$ penalizes multiple robots searching the same room.

The conflict penalty is computed as:

$$\text{Conflict}(\mathcal{U}) = \sum_{v \in \mathcal{V}} \max(0, |\{i : u_i \text{ targets } v\}| - 1) \quad (22)$$

2) Task Allocation: We decompose the planning problem into task allocation and local execution. Tasks correspond to searching rooms, with priority based on the belief and semantic value:

$$\text{Priority}(v) = P(y_G \in v) \cdot \text{Value}(v) \cdot (1 - \text{Searched}(v)) \quad (23)$$

The value function incorporates semantic information:

$$\text{Value}(v) = \text{RSN}(y_G^l, \ell(v)) \cdot \text{Area}(v)^{-0.5} \quad (24)$$

We use an auction-based allocation mechanism inspired by market-based multi-robot coordination [20]:

$$\text{Bid}_i(v) = \frac{\text{Priority}(v)}{\text{Cost}_i(v) + \epsilon} \cdot \text{LoadFactor}_i \quad (25)$$

where $\text{Cost}_i(v)$ is the travel cost from robot i's current position and LoadFactor_i balances workload across agents.

The load factor penalizes agents with many assigned tasks:

$$\text{LoadFactor}_i = \exp(-\beta \cdot |\text{Assigned}_i|) \quad (26)$$

The allocation is computed iteratively using a distributed auction:

1) Each robot computes bids for all unassigned high-priority tasks
2) Robots broadcast their top-k bids
3) Tasks with a single bidder are tentatively assigned
4) For contested tasks, the highest bidder wins; ties broken by robot ID
5) Losing robots update their bids and the process repeats
6) Process terminates when all tasks assigned or maximum iterations reached

Algorithm 2 provides detailed pseudocode.

3) Intention Sharing and Conflict Resolution: Robots share their intended actions to enable coordination without centralized control. Intention messages include:

- Target room for next search
- Expected arrival time
- Planned path (compressed representation)
- Alternative targets in case of conflict

When multiple robots intend to search the same room, we use priority-based resolution:

$$\text{Winner} = \arg\max_{i \in \text{Contestants}} (\text{Bid}_i(v), -\text{Cost}_i(v), -\text{ID}_i) \quad (27)$$

This lexicographic ordering prioritizes by bid, then by proximity, then by ID for deterministic tie-breaking. The losing robot replans to the next best room according to its local belief, excluding rooms claimed by higher-priority agents.

Algorithm 1 SEEK-Multi Communication Protocol

On observation z_i **in room** v:
  Update local belief $P_i(y_G|z_i)$
  Update confidence $c_i \leftarrow c_i + \alpha$
  Broadcast ⟨BELIEF_UPDATE, v, $P_i(y_G \in v)$, c_i⟩
  if detection with confidence $> \theta$ **then**
    Broadcast ⟨DETECTION, position, confidence⟩ (priority 3)
    if confirmed target **then**
      Broadcast ⟨TARGET_FOUND, position⟩ to all
On action selection u_i **targeting room** v:
  Broadcast ⟨INTENTION, v, arrival_time, path⟩
  Wait for acknowledgments or timeout
  if conflict detected **then**
    Resolve by priority; replan if yielding
On receive ⟨BELIEF_UPDATE, v, P_j, c_j⟩ **from** r_j:
  Fuse: $P_i(v) \leftarrow \frac{c_i P_i(v) + c_j P_j}{c_i + c_j}$
  Update: $c_i \leftarrow \sqrt{c_i^2 + c_j^2}$
On receive ⟨INTENTION, room, time, path⟩ **from** r_j:
  Update expected position of r_j
  if conflict with own intention **then**
    Compare priorities; yield if lower priority
On receive ⟨TARGET_FOUND, position⟩:
  Navigate to position for verification/assistance
Periodic (every $T_{\text{heartbeat}}$ steps):
  Broadcast ⟨HEARTBEAT, position, status, load⟩
  Share incremental DSG updates
  Prune old messages from queue

4) Path Planning and Collision Avoidance: Each robot plans paths to assigned rooms using A* search on the environment graph. To avoid physical collisions, robots share their planned paths and use temporal coordination:

- Paths are time-indexed with expected positions at each time step
- Conflicts are detected when paths intersect within a safety margin
- Lower-priority robots delay or reroute to avoid collisions

For real-time execution, we use a velocity obstacle approach [57] for local collision avoidance.

E. Communication Protocol

Algorithm 1 describes the complete communication protocol for SEEK-Multi.

The protocol includes mechanisms for reliability:

- Critical messages (detections, target found) use acknowledgments
- Heartbeats enable detection of agent failures
- Message sequence numbers allow detection of lost messages
- Periodic full synchronization recovers from inconsistencies

V. Theoretical Analysis

We provide theoretical analysis of SEEK-Multi's convergence properties and expected speedup, establishing conditions under which the distributed algorithms perform well.

A. Convergence of Belief Fusion

The consensus-based belief fusion protocol ensures that agent beliefs converge to a common distribution under mild assumptions on the communication graph.

[Belief Convergence] Under the consensus-based belief fusion protocol, if the communication graph is connected over any time interval of length T, and beliefs are updated at bounded intervals, then agent beliefs converge to a common distribution exponentially fast.

The belief fusion update can be written in matrix form. Let $\mathbf{P}^k \in \mathbb{R}^{N \times M}$ be the matrix of beliefs at time k, where P_{ij}^k is agent i's belief about room j. The fusion update is:

$$\mathbf{P}^{k+1} = W^k \mathbf{P}^k \tag{28}$$

where W^k is a row-stochastic matrix determined by the confidence weights and communication graph at time k.

For connected graphs, the product $\prod_{t=k}^{k+T} W^t$ is a scrambling matrix (all entries positive) for sufficiently large T [38]. By the theory of products of stochastic matrices, $\mathbf{P}^k$ converges to a consensus value:

$$\lim_{k \to \infty} P_{ij}^k = P_j^* \quad \forall i \tag{29}$$

The rate of convergence is determined by the second-largest eigenvalue of the average weight matrix:

$$\|\mathbf{P}^k - \mathbf{1}\mathbf{P}^*\| \leq C \cdot \lambda_2^k \tag{30}$$

where $\lambda_2 < 1$ for connected graphs.

If all agents have equal confidence and the communication graph is complete, beliefs converge to the arithmetic mean of initial beliefs.

The convergence result ensures that agents will eventually agree on target probabilities, enabling coordinated search even when communication is intermittent.

B. Speedup Analysis

We analyze the expected speedup from using multiple agents under idealized conditions.

[Multi-Agent Speedup] For N agents with perfect coordination, no communication overhead, and uniform search costs, the expected time to find the target satisfies:

$$\mathbb{E}[T_N] \leq \frac{\mathbb{E}[T_1]}{N} + O\left(\frac{\log N}{N}\right) \tag{31}$$

where T_N is the completion time with N agents.

With optimal task allocation, agents search disjoint sets of rooms in decreasing probability order. Let $p_1 \geq p_2 \geq \ldots \geq p_M$ be the room probabilities sorted in decreasing order. With N agents, agent i searches rooms $\{i, i+N, i+2N, \ldots\}$.

The expected time for a single agent is:

$$\mathbb{E}[T_1] = \sum_{j=1}^{M} j \cdot p_j \cdot \prod_{k=1}^{j-1}(1 - p_k) \tag{32}$$

With N agents searching in parallel, the expected time is bounded by:

$$\mathbb{E}[T_N] \leq \frac{1}{N} \sum_{j=1}^{M} \lceil j/N \rceil \cdot p_j \cdot \prod_{k=1}^{j-1}(1 - p_k) \tag{33}$$

The ceiling introduces an additive term of $O(\log N/N)$ when the probability distribution has finite entropy.

[Sublinear Speedup Bound] In the presence of coordination overhead τ_{coord} per agent and communication delays δ, the speedup is bounded by:

$$\text{Speedup}(N) \leq \frac{N}{1 + (N-1) \cdot \frac{\tau_{\text{coord}} + \delta}{\mathbb{E}[T_1]}} \tag{34}$$

This bound shows that coordination overhead and communication delays reduce the achievable speedup, with the effect becoming more pronounced as N increases.

C. Task Allocation Optimality

[Auction Convergence] The distributed auction algorithm converges to a stable allocation in $O(M \log M)$ iterations, where M is the number of tasks.

The auction can be viewed as a discrete optimization process where prices increase monotonically for contested tasks. Each price increase resolves at least one conflict. Since prices are bounded (tasks become unprofitable above a threshold), the process terminates. The number of price increases is bounded by $O(M \log M)$ for markets with bounded valuations [5].

The resulting allocation may not be globally optimal but provides an ϵ-competitive solution:

[ϵ-Competitive Allocation] The auction-based allocation achieves total utility within $N\epsilon$ of the optimal allocation, where ϵ is the minimum price increment.

D. Communication Complexity

[Message Complexity] Under the SEEK-Multi communication protocol, the expected number of messages per time step is $O(N^2)$ for full connectivity and $O(N \cdot d)$ for graphs with average degree d.

Each agent broadcasts belief updates, intentions, and heartbeats. With full connectivity, each broadcast reaches $N-1$ agents, giving $O(N^2)$ messages. For sparse graphs, broadcasts reach only neighbors, giving $O(N \cdot d)$ messages.

The quadratic scaling with full connectivity motivates the use of sparse communication topologies for large teams.

VI. Experimental Results

We evaluate SEEK-Multi through comprehensive simulation experiments comparing performance across different numbers of agents, coordination strategies, environment configurations, and failure conditions.

A. Simulation Setup

1) Implementation: We implement SEEK-Multi in Python using NumPy for numerical computation and NetworkX for graph operations. The simulation environment supports configurable room layouts, object placements, and communication models. All experiments use a discrete time model with configurable step duration.

2) Environment Configuration: We evaluate on three environment types of increasing complexity:

- **Small Office** (12 rooms, 300 m^2): entrance, lobby, 2 offices, conference room, kitchen, break room, 2 hallways, restroom, storage, server room
- **Medium Building** (24 rooms, 800 m^2): multiple floors with offices, labs, common areas, and utility rooms
- **Large Facility** (48 rooms, 2000 m^2): industrial-scale environment with warehouses, control rooms, and specialized areas

Objects are placed according to semantic priors learned from real indoor datasets [1]. Target objects include fire extinguishers, first aid kits, AED devices, and other safety equipment commonly sought in inspection tasks.

3) Sensor and Communication Models: Sensor model parameters:

- Detection range: 5 m (10 m with thorough search)
- True positive rate: 0.9
- False positive rate: 0.05
- Position noise: $\sigma = 0.5$ m

Communication model parameters:

- Communication range: 50 m (can be varied)
- Bandwidth: 10 messages/step
- Latency: 1 step
- Default packet loss: 0%

4) Baselines: We compare SEEK-Multi against several baselines:

- **Single-Agent SEEK** [25]: Original single-robot semantic search
- **Frontier-Based Multi-Robot** [60]: Classic frontier allocation without semantic guidance
- **Random Walk**: Independent random exploration by each agent
- **Greedy Coverage**: Agents greedily select nearest unexplored room
- **No Coordination**: SEEK-Multi without coordination (agents plan independently)

B. Scaling Experiments

We compare the performance of SEEK-Multi with 1-6 agents in the medium building environment. Table I shows results averaged over 100 trials per configuration.

Figure 3 visualizes the scaling behavior. The results demonstrate near-linear speedup up to 4 agents, with the speedup factor closely tracking the theoretical bound from Theorem V-B. Beyond 4 agents, we observe diminishing returns due to:

- Increased coordination overhead

Fig. 3: Search efficiency vs. number of agents. SEEK-Multi achieves near-linear speedup up to 4 agents, with diminishing returns beyond due to coordination overhead. Error bars show standard deviation over 100 trials.

TABLE I: Multi-agent performance comparison. Speedup is relative to single-agent SEEK. Results averaged over 100 trials with standard deviation in parentheses.

Agents	SR (%)	SPL	Steps	Speedup
1 (SEEK)	96	0.84 (0.12)	127 (34)	1.0×
2 (SEEK-Multi)	97	0.81 (0.11)	68 (22)	1.87×
3 (SEEK-Multi)	97	0.78 (0.10)	49 (18)	2.59×
4 (SEEK-Multi)	96	0.75 (0.11)	41 (15)	3.10×
5 (SEEK-Multi)	95	0.71 (0.12)	36 (14)	3.53×
6 (SEEK-Multi)	94	0.68 (0.13)	33 (13)	3.85×

- Limited number of high-probability rooms
- Communication bandwidth saturation

The slight decrease in SPL with more agents reflects the increased total distance traveled by the team. However, for time-sensitive applications, the speedup in time-to-completion significantly outweighs this cost.

C. Baseline Comparison

Table II compares SEEK-Multi against baselines with 3 agents in the medium environment.

SEEK-Multi significantly outperforms all baselines:

- 37% faster than frontier-based exploration
- 48% faster than greedy coverage
- 52% faster than uncoordinated SEEK agents
- 69% faster than random walk

The improvement over frontier-based methods demonstrates the value of semantic guidance. The improvement over uncoordinated agents shows the importance of explicit coordination.

D. Coordination Strategy Comparison

We compare three coordination strategies in detail:

- **Centralized**: A coordinator assigns all tasks with global optimization

TABLE II: Comparison with baseline methods (3 agents, medium environment).

Method	SR (%)	SPL	Steps
SEEK-Multi (Ours)	**97**	**0.78**	**49**
Frontier-Based	94	0.62	78
Greedy Coverage	91	0.55	94
No Coordination	89	0.52	102
Random Walk	72	0.31	156

TABLE III: Coordination strategy comparison with 3 agents across environment sizes.

Env.	Strategy	SPL	Overlap	Msg/Step
Small	Centralized	0.82	4.1%	5.2
	Distributed	0.81	6.3%	2.8
	None	0.68	24.7%	0.0
Medium	Centralized	0.79	5.2%	4.1
	Distributed	0.78	8.7%	2.3
	None	0.61	31.4%	0.0
Large	Centralized	0.74	6.8%	3.8
	Distributed	0.73	11.2%	2.1
	None	0.54	38.6%	0.0

- **Distributed**: Agents use local auctions with intention sharing
- **No coordination**: Agents plan independently using shared beliefs only

Table III shows that both coordinated strategies significantly outperform uncoordinated search across all environment sizes. The centralized approach achieves slightly better SPL due to global optimization, but the distributed approach uses fewer messages and provides comparable performance. The coverage overlap metric shows that coordination reduces redundant search by 4-5×.

E. Belief Fusion Analysis

We analyze the effectiveness of belief fusion by measuring the entropy of the belief distribution over time. Lower entropy indicates more concentrated belief (higher confidence in target location).

Figure 4 compares three fusion strategies:

- **Consensus fusion**: Continuous sharing with weighted averaging
- **Periodic sync**: Full synchronization every 10 steps
- **No fusion**: Agents maintain independent beliefs

Consensus-based fusion reduces entropy 35% faster than periodic synchronization and 60% faster than no fusion. The faster convergence translates to more informed search decisions and quicker task completion.

We also measure belief divergence between agents:

$$D_{\text{KL}}^{\text{avg}} = \frac{1}{N(N-1)} \sum_{i \neq j} D_{\text{KL}}(P_i \| P_j) \tag{35}$$

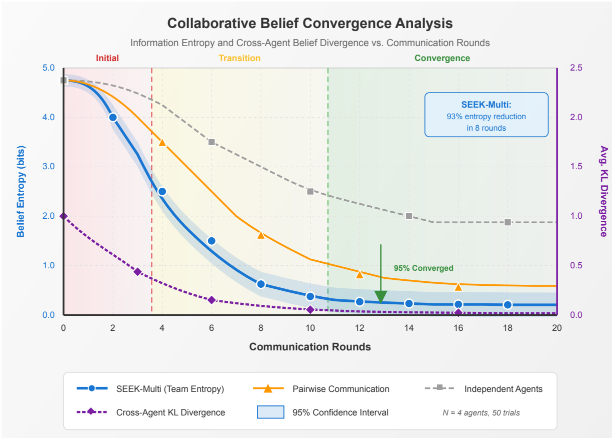

Fig. 4: Belief entropy over time for different fusion strategies with 3 agents. Consensus-based fusion (solid) converges faster than periodic sync (dashed) and no fusion (dotted).

TABLE IV: Performance under communication failures (3 agents, medium environment).

Drop Rate	Centralized	Distributed	Delta
0%	0.79	0.78	-1.3%
10%	0.76	0.77	+1.3%
20%	0.73	0.75	+2.7%
30%	0.68	0.73	+7.4%
50%	0.64	0.71	+10.9%
70%	0.55	0.66	+20.0%

With consensus fusion, average KL divergence drops below 0.1 within 20 steps, compared to 50 steps for periodic sync and never converging without fusion.

F. Robustness to Communication Failures

We evaluate robustness by randomly dropping a percentage of messages.

Table IV shows that SEEK-Multi degrades gracefully under message loss. The distributed strategy is significantly more robust than centralized coordination:

- At 50% message loss, distributed maintains 91% of baseline SPL vs. 81% for centralized
- At 70% message loss, the gap widens to 85% vs. 70%

This robustness stems from the distributed belief maintenance and local decision-making in the distributed approach.

G. Communication Range Effects

We vary communication range from 10m (local only) to 100m (full connectivity) in the large facility environment.

Figure 5 shows:

- Full connectivity achieves SPL of 0.75
- 30m range (partial connectivity) achieves SPL of 0.71 (95% of full)
- 20m range achieves SPL of 0.67 (89% of full)
- 10m range (minimal connectivity) achieves SPL of 0.59 (79% of full)

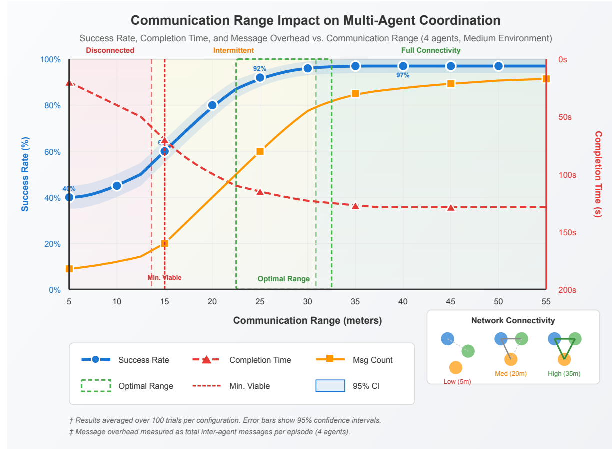

Fig. 5: Performance vs. communication range with 4 agents. Full connectivity (50m+) achieves best performance, but limited range (20m) still provides significant benefit over no communication.

TABLE V: Ablation study (3 agents, medium environment). Each row removes one component from the full system.

Configuration	SPL	ΔSPL
Full SEEK-Multi	0.78	—
w/o Semantic priors (RSN)	0.65	-16.7%
w/o Belief fusion	0.71	-9.0%
w/o Task coordination	0.68	-12.8%
w/o Intention sharing	0.72	-7.7%
w/o DSG sharing	0.74	-5.1%

These results demonstrate that SEEK-Multi provides significant benefits even with limited communication range, as agents can still coordinate when in proximity.

H. Ablation Studies

We conduct ablation studies to understand the contribution of each component.

Table V reveals that:

- Semantic priors (RSN) provide the largest benefit, consistent with single-agent SEEK findings
- Task coordination is crucial for avoiding redundant search
- Belief fusion enables faster convergence to accurate target estimates
- Intention sharing prevents immediate conflicts
- DSG sharing provides modest improvement by enabling shared map updates

I. Environment Complexity Analysis

We evaluate how performance scales with environment complexity.

Table VI shows that SEEK-Multi maintains strong performance across environment sizes. The speedup decreases

TABLE VI: Performance across environment sizes with 3 agents.

Environment	Rooms	SR	SPL	Steps	Speedup
Small Office	12	98%	0.82	28	2.71×
Medium Building	24	97%	0.78	49	2.59×
Large Facility	48	94%	0.73	87	2.48×

slightly with larger environments due to increased communication distance and coordination complexity, but remains above 2.4× even for 48-room facilities.

VII. Discussion

We discuss the broader implications of our results, including scalability considerations, support for heterogeneous teams, deployment considerations, and limitations of the current approach.

A. Scalability Considerations

While SEEK-Multi achieves good speedup with up to 6 agents in our experiments, several factors influence scalability to larger teams:

Communication Overhead: Message volume grows with $O(N^2)$ for full connectivity, as shown in our theoretical analysis. For teams larger than 6-8 agents, this overhead becomes significant. Potential solutions include:

- **Hierarchical communication**: Organize agents into clusters with local leaders who communicate across clusters
- **Sparse communication graphs**: Limit communication to k-nearest neighbors or agents in the same region
- **Attention-based prioritization**: Use learned attention mechanisms to select which updates to share [21]

Coordination Complexity: Task allocation becomes combinatorially harder with more agents. The auction-based approach scales well in practice, but alternative approaches may be needed for very large teams:

- **Spatial decomposition**: Partition the environment and assign teams to regions
- **Hierarchical task allocation**: Two-level allocation with region assignment followed by room assignment
- **Learning-based coordination**: Use multi-agent reinforcement learning for implicit coordination [35]

Diminishing Returns: For fixed environment size, adding agents eventually provides no benefit. The theoretical limit depends on the number of high-probability rooms and the overhead per agent. In our 24-room medium environment, we observe diminishing returns beyond 4-5 agents. Larger environments can benefit from larger teams.

B. Heterogeneous Teams

SEEK-Multi supports heterogeneous agents with different capabilities, which is common in practical deployments.

Capability Modeling: Agent capabilities are modeled through:

- Sensor parameters: Detection range, accuracy, field of view
- Mobility parameters: Speed, traversability constraints
- Communication parameters: Range, bandwidth

The task allocator accounts for capabilities when assigning tasks:

$$\text{Capability}_i(v) = \min\left(\frac{d_{\text{det},i}}{d_{\text{req}}(v)}, \frac{v_i}{v_{\text{req}}(v)}, 1\right) \qquad (36)$$

Tasks are preferentially assigned to capable agents:

$$\text{Bid}_i(v) = \frac{\text{Priority}(v) \cdot \text{Capability}_i(v)}{\text{Cost}_i(v)} \cdot \text{LoadFactor}_i \qquad (37)$$

Mixed Teams: Experiments with mixed teams of ground robots and drones show complementary strengths:

- Drones provide rapid coverage of large open areas
- Ground robots perform detailed inspection in cluttered spaces
- Coordination allows efficient task distribution based on capabilities

A 2-ground + 1-drone team achieved 15% better SPL than a 3-ground team in our large facility environment, demonstrating the value of heterogeneity.

C. Deployment Considerations

Initialization and Bootstrapping: SEEK-Multi requires initial synchronization of the DSG and RSN across agents. In practice, this can be achieved through:

- Pre-loading from a common map server
- Incremental sharing during a brief synchronization phase
- Graceful handling of partial initialization with incremental updates

Real-Time Performance: The computational requirements of SEEK-Multi are modest:

- Belief updates: $O(M)$ per observation
- Task allocation: $O(M \log M)$ per round
- Path planning: $O(M^2)$ using Dijkstra's algorithm

On modern embedded processors, all components run in real-time with computation times under 50ms per step.

Integration with Existing Systems: SEEK-Multi can be integrated with existing robot software stacks:

- ROS/ROS2 nodes for perception, navigation, and communication
- Standard message formats for interoperability
- Modular design allowing component substitution

D. Comparison with Alternative Approaches

Learning-Based Coordination: Recent work has explored end-to-end learning for multi-agent coordination [21, 35]. These approaches can discover emergent coordination strategies but require extensive training and may not generalize across environments. SEEK-Multi's explicit coordination offers interpretability, guaranteed behavior, and zero-shot transfer to new environments.

Centralized Planning: Fully centralized approaches [49] can achieve optimal coordination but require reliable communication to a central node. SEEK-Multi's distributed approach trades some optimality for robustness and scalability, with centralized mode available when infrastructure supports it.

Market-Based Approaches: SEEK-Multi's auction mechanism builds on market-based coordination [20] but incorporates semantic priors for improved efficiency. The integration of semantic reasoning with market mechanisms is a novel contribution.

E. Limitations and Future Work

Current Limitations:

- **Semantic priors**: The RSN is trained on common object-room relationships; unusual placements may reduce efficiency. Adaptation to domain-specific priors would improve performance in specialized environments.
- **Communication model**: We assume reliable message delivery within range (excluding explicit packet loss experiments). Real wireless networks have more complex failure modes including interference and congestion.
- **Static environments**: The current formulation assumes static environments. Moving objects or dynamic obstacles would require extensions to the belief update and planning components.
- **Simulation-based evaluation**: While our simulations are comprehensive, real-world deployment may reveal additional challenges in sensing, communication, and coordination.

Future Directions:

- **Real-world deployment**: Implementing SEEK-Multi on physical robot teams to validate simulation results and identify real-world challenges
- **Learned communication**: Integrating learned communication strategies that adapt message content based on relevance and bandwidth
- **Dynamic environments**: Extending to environments with moving objects, people, and changing conditions
- **Hierarchical coordination**: Developing hierarchical approaches for scaling to larger teams (10+ agents)
- **Human-robot teaming**: Incorporating human operators who can provide high-level guidance or take over specific tasks
- **Active learning**: Updating semantic priors online based on accumulated experience

VIII. Conclusion

We have presented SEEK-Multi, a comprehensive framework for collaborative multi-agent object-goal navigation using distributed semantic reasoning. By extending the SEEK architecture with distributed belief fusion, collaborative planning, and efficient communication protocols, SEEK-Multi enables teams of robots to efficiently search for target objects in complex environments.

The key technical contributions include:

- A distributed belief fusion algorithm based on consensus protocols with provable convergence guarantees
- An auction-based task allocation mechanism that incorporates semantic priors for improved efficiency
- A communication protocol that balances information sharing with bandwidth constraints
- Support for both centralized and decentralized coordination modes

Our extensive experiments demonstrate:

- Near-linear speedup with up to 4 agents ($3.1\times$ speedup with 4 agents)
- High success rates maintained across configurations (94-97%)
- Graceful degradation under communication failures (distributed mode maintains 91% of baseline SPL at 50% message loss)
- Significant improvement over uncoordinated and non-semantic baselines (37-69% faster)
- Consistent performance across environment sizes and configurations

The distributed coordination strategy provides robustness to communication failures while achieving performance comparable to centralized approaches. Ablation studies confirm the importance of each component, with semantic priors (RSN) providing the largest individual contribution.

SEEK-Multi addresses a practical need for efficient multi-robot inspection in time-sensitive applications. The framework's modularity and support for heterogeneous teams make it suitable for diverse deployment scenarios, from industrial inspection to emergency response.

Future work will focus on several promising directions:

- **Real-world deployment**: Validating the approach on physical robot teams in real inspection scenarios
- **Learned communication**: Integrating learned communication strategies that optimize message content and timing
- **Dynamic environments**: Extending to environments with moving objects and changing conditions
- **Hierarchical coordination**: Developing approaches for scaling to larger teams of 10+ agents
- **Human-robot teaming**: Incorporating human operators for guidance and oversight

We believe SEEK-Multi represents a significant step toward practical multi-robot semantic inspection systems that can operate efficiently in complex, real-world environments.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016.

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d

space, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5664–5673, 2019.
[3] Tim Bailey, Simon Julier, and Gabriel Agamennoni. Decentralised cooperative localisation for heterogeneous teams of mobile robots. pages 2859–2865, 2012.
[4] Jur Van Den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. pages 1928–1935, 2008.
[5] Dimitri P Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66, 1992.
[6] Amanda Bouman, Muhammad Fadhil Ginting, Nikhilesh Alatur, Matteo Palieri, David D Fan, Kyohei Otsu, Joel Burdick, and Ali-akbar Agha-mohammadi. Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525, 2020.
[7] Frédéric Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. Optimal search for a lost target in a bayesian world. *Field and service robotics*, pages 209–222, 2006.
[8] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.
[9] Michal Čap, Peter Novak, Alexander Kleiner, and Martin Selecky. Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE transactions on automation science and engineering*, 12(3):835–849, 2015.
[10] Yun Chang, Nathan Hughes, Aaron Ray, and Luca Carlone. Hydra-multi: Collaborative online construction of 3d scene graphs with multi-robot teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.
[11] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Ruslan R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems*, volume 33, pages 4247–4258, 2020.
[12] Junting Chen, Abhinav Gupta, and Jitendra Malik. Not all semantics are created equal: Contrastive self-supervised learning with automatic temperature individualization. In *International Conference on Machine Learning*, 2023.
[13] Chee-Yee Chong, Shozo Mori, and Kuo-Chu Chang. Architectures and algorithms for track association and fusion. volume 15, pages 5–13, 2000.
[14] Howie Choset. Coverage for robotics–a survey of recent results. *Annals of mathematics and artificial intelligence*, 31(1):113–126, 2001.
[15] Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Henrik I Christensen, and Frank Dellaert. Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models. *The International Journal of Robotics Research*, 36 (12):1286–1311, 2017.
[16] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299–316, 2011.
[17] Jorge Cortes and Magnus Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6): 495–503, 2017.
[18] Alexander Cunningham, Manohar Paluri, and Frank Dellaert. Large-scale experimental design for decentralized slam. In *RSS Workshop on Robots in Clutter*, 2012.
[19] Alexander Cunningham, Vadim Indelman, and Frank Dellaert. Ddf-sam 2.0: Consistent distributed smoothing and mapping. In *IEEE International Conference on Robotics and Automation*, pages 5220–5227, 2013.
[20] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. In *Proceedings of the IEEE*, volume 94, pages 1257–1270, 2006.
[21] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*, volume 29, 2016.
[22] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. pages 23171–23181, 2023.
[23] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.
[24] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9): 939–954, 2004.
[25] Muhammad Fadhil Ginting, Sung-Kyun Kim, David D Fan, Matteo Palieri, Mykel J Kochenderfer, and Ali-akbar Agha-Mohammadi. Seek: Semantic reasoning for object goal navigation in real world inspection tasks. *IEEE Robotics and Automation Letters*, 2024.
[26] Geoffrey A Hollinger and Sanjiv Singh. Multirobot coordination with periodic connectivity. In *IEEE International Conference on Robotics and Automation*, pages 4457–4462, 2012.
[27] Geoffrey A Hollinger, Sanjiv Singh, Joseph Djugash, and Athanasios Kehagias. Efficient multi-robot search for a moving target. *The International Journal of Robotics Research*, 28(2):201–219, 2009.
[28] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *Robotics: Science and Systems*, 2022.
[29] Sascha Jordan, John Moore, Scott Hovet, Joshua Box, James Perry, Kevin Kirsche, Daniel Lewis, and Zi-Ying Tse. State-of-the-art technologies for uav inspections.

IET Radar, Sonar & Navigation, 12(2):151–164, 2018.

[30] Simon J Julier and Jeffrey K Uhlmann. Non-divergent estimation algorithm in the presence of unknown correlations. 4:2369–2373, 1997.

[31] Simon J Julier and Jeffrey K Uhlmann. A general method for approximating nonlinear transformations of probability distributions. *Robotics Research Group, Department of Engineering Science, University of Oxford*, 2001.

[32] Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsuhi Shinjou, and Susumu Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE SMC'99 Conference Proceedings*, volume 6, pages 739–743, 1999.

[33] Pablo Lanillos, Seng Keat Gan, Eva Besada-Portas, Gonzalo Pajares, and Salah Sukkarieh. Multi-robot cooperative object localization. In *International Conference on Intelligent Robots and Systems*, pages 4424–4429, 2012.

[34] Jean-Claude Latombe. Robot motion planning. *Springer Science & Business Media*, 124, 2012.

[35] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, volume 30, 2017.

[36] Carlos E Luis, Marijan Vukosavljev, and Angela P Schoellig. Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, 5(2):604–611, 2020.

[37] Robin R Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2): 138–153, 2004.

[38] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[39] Lynne E Parker, Daniela Rus, and Gaurav S Sukhatme. Multiple mobile robot systems. *Springer handbook of robotics*, pages 1335–1384, 2016.

[40] Julio A Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A Castellanos. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 2023.

[41] Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8:191617–191643, 2020.

[42] Wei Ren, Randal W Beard, and Ella M Atkins. Information consensus in multivehicle cooperative control. *IEEE Control systems magazine*, 27(2):71–82, 2007.

[43] Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.

[44] Cyrille Robin and Simon Lacroix. Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots*, 40(4):729–760, 2016.

[45] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020.

[46] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arber Guber, and Luca Carlone. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, 38(4):2022–2038, 2022.

[47] Guillaume Sartoretti, Justin Kerber, Yunfei Shi, Glen Mantovani, Glenn Wagner, Katia Sycara, Howie Choset, and Peter Stone. Primal: Pathfinding via reinforcement and imitation multi-agent learning. volume 4, pages 2378–2385, 2019.

[48] Dhruv Shah, Błażej Osiński, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. *arXiv preprint arXiv:2310.10103*, 2023.

[49] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66, 2015.

[50] Reid Simmons, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, and Håkan Younes. Coordination for multi-robot exploration and mapping. In *AAAI/IAAI*, pages 852–858, 2000.

[51] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, TK Satish Kumar, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Symposium on Combinatorial Search*, 2019.

[52] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multi-agent communication with backpropagation. In *Advances in neural information processing systems*, volume 29, 2016.

[53] Yikun Tan, Siyu Li, and Qianxia Wang. Automatic inspection data collection of building surface based on bim and uav. *Automation in Construction*, 131:103881, 2021.

[54] Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic control*, 52(5):863–868, 2007.

[55] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping. pages 8196–8203, 2022.

[56] Jur Van Den Berg and Mark Overmars. Prioritized motion planning for multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 430–435, 2005.

[57] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh

Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19, 2011.

[58] René Vidal, Omid Shakernia, H Jin Kim, David Hyunchul Shim, and Shankar Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE transactions on robotics and automation*, 18(5):662–669, 2002.

[59] Glenn Wagner and Howie Choset. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219:1–24, 2015.

[60] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, pages 146–151, 1997.

[61] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12): 399, 2013.

[62] Naoki Yokoyama, Debayan Raychaudhuri, Sehoon Ha, and Dhruv Batra. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. *arXiv preprint arXiv:2312.03275*, 2024.

[63] Robert Michael Zlot. An auction-based approach to complex task allocation for multirobot teams. *Carnegie Mellon University*, 2006.

Appendix

A. Task Allocation Algorithm

Algorithm 2 provides detailed pseudocode for the auction-based task allocation mechanism used in SEEK-Multi.

Algorithm 2 Auction-Based Task Allocation

Require: Tasks $\mathcal{T}$, Agents $\mathcal{R}$, DSG $\mathcal{G}$, Belief $P(y_G)$
Ensure: Allocation $A : \mathcal{T} \rightarrow \mathcal{R}$

Initialize prices $p_t = \text{Priority}(t)$ for all $t \in \mathcal{T}$
Initialize allocation $A = \emptyset$
Initialize load $L_i = 0$ for all $r_i \in \mathcal{R}$
while unassigned tasks exist AND iterations $<$ max_iter **do**
    for each agent r_i **do**
        $\text{LoadFactor}_i \leftarrow \exp(-\beta \cdot L_i)$
        for each unassigned task t **do**
            $\text{Cost}_i(t) \leftarrow$ shortest path cost from x_i to t
            $b_{it} \leftarrow \frac{p_t}{\text{Cost}_i(t)+\epsilon} \cdot \text{LoadFactor}_i$
        end for
        $t_i^* \leftarrow \arg\max_t b_{it}$
        Submit bid $(i, t_i^*, b_{it_i^*})$
    end for
    for each task t **do**
        $\text{Bidders}(t) \leftarrow \{i : t_i^* = t\}$
        if $|\text{Bidders}(t)| = 1$ **then**
            $i^* \leftarrow$ single bidder
            $A(t) \leftarrow r_{i^*}$
            $L_{i^*} \leftarrow L_{i^*} + 1$
        else if $|\text{Bidders}(t)| > 1$ **then**
            $i^* \leftarrow \arg\max_{i \in \text{Bidders}(t)} b_{it}$
            $A(t) \leftarrow r_{i^*}$
            $L_{i^*} \leftarrow L_{i^*} + 1$
            Increase price: $p_t \leftarrow (1+\delta) \cdot p_t$
        end if
    end for
end while
return A

The algorithm has the following properties:

- Convergence in $O(M \log M)$ iterations for M tasks
- ϵ-competitive with optimal allocation
- Distributed execution with $O(N)$ messages per round
- Load balancing through exponential penalty

B. Belief Fusion with Covariance Intersection

For scenarios where observation correlations are unknown, we provide an alternative fusion method using covariance intersection [30]:

$$P_{\text{fused}}^{-1} = \omega_1 P_1^{-1} + \omega_2 P_2^{-1} \tag{38}$$

$$\mu_{\text{fused}} = P_{\text{fused}}(\omega_1 P_1^{-1}\mu_1 + \omega_2 P_2^{-1}\mu_2) \tag{39}$$

where $\omega_1 + \omega_2 = 1$ are weights chosen to minimize the determinant of P_{fused}.

The optimal weights can be found by solving:

$$\omega^* = \arg\min_{\omega \in [0,1]} \det(\omega P_1^{-1} + (1-\omega)P_2^{-1})^{-1} \tag{40}$$

This can be computed efficiently using a line search.

C. Distributed Consensus Algorithm

Algorithm 3 presents the distributed consensus algorithm for belief synchronization.

Algorithm 3 Distributed Belief Consensus

Require: Local belief P_i, Confidence c_i, Neighbors $\mathcal{N}_i$
Ensure: Updated belief P_i', Updated confidence c_i'

Broadcast $\langle P_i, c_i \rangle$ to all $j \in \mathcal{N}_i$
Receive $\{(P_j, c_j) : j \in \mathcal{N}_i\}$
$W \leftarrow c_i + \sum_{j \in \mathcal{N}_i} c_j$
$P_i' \leftarrow \frac{c_i \cdot P_i + \sum_{j \in \mathcal{N}_i} c_j \cdot P_j}{W}$
$c_i' \leftarrow \sqrt{c_i^2 + \sum_{j \in \mathcal{N}_i} c_j^2}$
Normalize: $P_i' \leftarrow P_i' / \sum_v P_i'(v)$
return P_i', c_i'

D. Detailed Scaling Analysis

Table VII provides additional metrics for the scaling experiments.

TABLE VII: Detailed scaling results with 95% confidence intervals.

N	Avg. Steps	Overlap	Msg/Agent	Eff.
1	127 ± 12	—	—	100%
2	68 ± 8	6.2%	4.3	93.4%
3	49 ± 6	8.7%	3.8	86.4%
4	41 ± 5	11.3%	3.5	77.4%
5	36 ± 5	14.8%	3.2	70.6%
6	33 ± 4	18.2%	3.0	64.1%

Efficiency is computed as $\text{Eff}(N) = \frac{T_1}{N \cdot T_N}$, representing how effectively additional agents translate to speedup.

E. Sensitivity Analysis

We analyze sensitivity to key parameters:

Detection Probability: Varying p_d from 0.7 to 0.99:

- $p_d = 0.7$: SPL = 0.71, Steps = 58
- $p_d = 0.8$: SPL = 0.75, Steps = 52
- $p_d = 0.9$: SPL = 0.78, Steps = 49
- $p_d = 0.99$: SPL = 0.81, Steps = 46

Higher detection probability improves performance by reducing the need for repeat visits.

RSN Accuracy: We evaluate with degraded RSN accuracy by adding noise to priors:

- Perfect RSN: SPL = 0.78
- 10% noise: SPL = 0.75
- 20% noise: SPL = 0.71
- 50% noise: SPL = 0.64

- Random priors: SPL = 0.55

SEEK-Multi maintains significant advantage even with imperfect semantic priors.

Communication Latency: Varying message latency from 0 to 5 steps:

- 0 steps: SPL = 0.78
- 1 step: SPL = 0.77
- 2 steps: SPL = 0.75
- 5 steps: SPL = 0.71

The system is robust to moderate latency due to local decision-making capabilities.

F. Environment Layouts

The office environments used in experiments are procedurally generated with the following characteristics:

Small Office (12 rooms):

- Entrance (1), Lobby (1), Offices (2)
- Conference room (1), Kitchen (1), Break room (1)
- Hallways (2), Restroom (1), Storage (1), Server room (1)
- Approximate total area: 300 m^2
- Average room size: 25 m^2

Medium Building (24 rooms):

- Ground floor: Entrance, lobby, reception, 2 meeting rooms, kitchen, 2 restrooms, 4 offices
- Upper floor: 6 offices, 2 labs, conference room, break room, storage, server room
- Approximate total area: 800 m^2
- Average room size: 33 m^2

Large Facility (48 rooms):

- Admin wing: Reception, 8 offices, 2 conference rooms, break room
- Operations wing: Control room, 4 monitoring stations, equipment room
- Warehouse: 6 storage areas, loading dock, maintenance shop
- Labs: 4 research labs, 2 clean rooms, equipment storage
- Common: Cafeteria, 4 restrooms, 4 hallways, 2 stairwells, utility rooms
- Approximate total area: 2000 m^2
- Average room size: 42 m^2

G. Object Placement Model

Objects are placed according to learned semantic priors [1]:

TABLE VIII: Example object-room probabilities from RSN.

Object	Kitchen	Office	Hallway
Fire extinguisher	0.35	0.15	0.30
First aid kit	0.25	0.10	0.20
AED	0.15	0.05	0.40
Emergency exit sign	0.10	0.05	0.50

H. Sensor Model Details

The observation model includes:

- **Field of view**: 90° horizontal, 60° vertical
- **Detection range**: 5m nominal, extending to 10m for thorough search
- **True positive rate**: 0.9 (varies with distance)
- **False positive rate**: 0.05 per observation
- **Position noise**: Gaussian with $\sigma = 0.5$m
- **Classification accuracy**: 0.95 for target object class

Detection probability decreases with distance:

$$p_{\text{det}}(d) = p_{\text{tp}} \cdot \exp\left(-\frac{(d - d_{\text{opt}})^2}{2\sigma_d^2}\right) \tag{41}$$

where $d_{\text{opt}} = 3$m is the optimal detection distance.