



**ARTICLE**

# Addressing Uncertainties in Decentralized Context Models of Autonomous Robot Teams

Marvin Zager<sup>1,\*</sup>, Gianluca Manca<sup>2</sup>, Alexander Fay<sup>2</sup> and Felix Gehlhoff<sup>1</sup>

<sup>1</sup>Institute of Automation Technology, Helmut Schmidt University, Holstenhofweg 85, Hamburg, Germany

<sup>2</sup>Chair of Automation, Ruhr University, Universitätsstraße 150, Bochum, Germany

\*Corresponding Author: Marvin Zager. Email: [marvin.zager@hsu.hamburg](mailto:marvin.zager@hsu.hamburg)

Received: 13 January 2026; Accepted: 08 March 2026; Published: 27 April 2026

**ABSTRACT:** Autonomous robot teams operating in dynamic, uncertain environments require reliable mechanisms to build decentralized context models without centralized coordination. Traditional consensus methods often fail under uncertainty caused by inconsistent sensing, communication delays, or heterogeneous perception models. This paper introduces the Decentralized Belief Consensus (DBC) algorithm, a novel approach that integrates probabilistic reasoning with entropy-based certainty measures to enable adaptive and robust consensus formation in heterogeneous multi-robot systems. Each robot quantifies the uncertainty of its local observations using Shannon entropy, derives a certainty score, and fuses beliefs with neighbors through certainty-weighted averaging. This allows the team of autonomous robots to defer commitment when evidence is weak and dynamically adjust influence according to observation reliability. The DBC algorithm was evaluated through various simulations involving heterogeneous teams of unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) tasked with mine detection under varying levels of noise, false detections, and team sizes. Results demonstrate that DBC achieves high accuracy, full consensus rates, and strong robustness while maintaining competitive convergence times compared to established algorithms such as LCP, WMSR, CDCI, DBBS, and EEV. By explicitly modeling uncertainty in both sensing and communication, DBC provides a scalable foundation for reliable decentralized context modeling and collective perception in autonomous robot teams.

**KEYWORDS:** Autonomous robots; context model; decentralized consensus; uncertainty

## 1 Introduction

### 1.1 Autonomous Robot Teams

Autonomous robots are increasingly deployed in missions that are dangerous, complex, or beyond the endurance of human operators. Examples include search and rescue operations after natural disasters [1], the detection of hazardous materials [2], or planetary exploration [3]. In such contexts, speed, robustness, and reliability are decisive. A single robot, however, is rarely sufficient to achieve these objectives. The limitations in sensing range, mobility, endurance, or other capabilities of an individual robotic system necessitate the cooperation of multiple robots to form a capable team [4].

Heterogeneous robot teams have emerged as an effective approach to address the constraints of a singular robot. By combining different modalities such as unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs), and unmanned surface vehicles (USVs), teams can exploit complementary capabilities [4,5]. For example, UAVs provide high mobility and wide-area reconnaissance, while UGVs

deliver precise sensing at close range. Moreover, failures of individual robots can be compensated by others, improving overall resilience [4].

The cooperation of autonomous robots requires effective coordination and communication. Two paradigms exist: centralized coordination, in which a central node assigns tasks and collects information, and decentralized coordination, in which robots exchange data directly and refine their local world models iteratively [6]. While centralized approaches may simplify coordination, they introduce a single point of failure and require high communication bandwidth. In contrast, decentralized approaches offer robustness, scalability, and flexibility, which makes them more appropriate for dynamic environments [7,8]. However, decentralized systems also face inherent challenges. Since each robot maintains only a local and often incomplete model of the environment and its context, collective decisions may be based on partial or outdated information. This drawback must be considered in the design of decentralized context models and coordination mechanisms.

### ***1.2 Decentralized Context Modeling and Consensus***

For a team of autonomous robots to cooperate effectively, they must maintain a shared understanding of their operational environment; otherwise, the robots risk converging on incomplete, inconsistent, or even incorrect information [7]. This shared understanding is captured in a context model, which represents relevant environmental entities, such as objects of interest, their positions, types, and associated confidence values. Each robot contributes local observations that are inherently limited and often inconsistent. By exchanging such information within the robot team, a decentralized context model can be formed that integrates multiple viewpoints into a more complete and reliable representation of the environment [7,8].

Forming a decentralized context model requires mechanisms for consensus. In this setting, consensus refers to the alignment of distributed beliefs across the robot team, ensuring that all members converge toward a consistent interpretation of their environment [7]. Unlike centralized coordination, where a central authority aggregates and redistributes information, decentralized consensus relies exclusively on peer-to-peer communication and local updates. This eliminates single points of failure and enhances scalability, but places higher demands on the robustness of the underlying algorithms [8,9].

Consensus mechanisms have been extensively studied in distributed systems and multi-robot coordination. [Section 3](#) provides an overview of the most relevant related work in the domain of consensus formation within decentralized robot teams.

### ***1.3 Uncertainty and the Research Gap***

Operating in real-world environments exposes autonomous robot teams to various forms of uncertainty. Even in cooperative settings where all agents are trustworthy, each robot perceives only a partial and often inconsistent view of its surroundings. Observations differ in consistency, reliability, and timing depending on the sensing modality, platform dynamics, and environmental conditions [9,10]. As a result, the information exchanged within the team can be uncertain, asynchronous, and sometimes contradictory.

Uncertainty arises at several levels. Accurate localization itself remains a fundamental challenge in mobile robotics, particularly under dynamic and GPS-denied conditions [11]. At the sensor level, measurement noise, occlusion, and heterogeneous sensing fidelity lead to varying data quality, for instance between UAVs providing wide-area but coarse observations and UGVs delivering close-range but localized measurements [12]. At the communication level, message delays, packet loss, and asynchronous transmission lead to outdated or inconsistent information [13]. At the agent level, temporary failures, intermittent participation, and dynamic team composition further complicate consensus formation [14]. Finally, at the

environmental level, changing terrain, weather conditions, or moving objects affect the robots' perception and decision-making processes [15]. While these forms of uncertainty have been widely studied, classical consensus mechanisms often assume homogeneous reliability or rely on fixed aggregation rules. When evidence is weak, delayed, or inconsistent, such methods may converge prematurely, fail to converge, or oscillate between conflicting states [6].

Beyond the above-mentioned sources of uncertainty, algorithmic uncertainty is gaining importance. Depending on the task, differently calibrated or trained machine learning models for perception can introduce additional discrepancies. Variations in network architecture, training data, or calibration parameters may lead to divergent posterior class probability distributions across robots observing the same object (e.g., ensemble methods in vision systems show model disagreement) [16]. Unlike sensor or communication noise, this form of uncertainty originates from differences in the internal reasoning of perception systems. It becomes particularly relevant as teams of heterogeneous robots, which might be equipped with distinct hardware, software versions, or proprietary AI models from different vendors, collaborate in the same mission. Even when confronted with identical real-world stimuli, such systems may exhibit varying confidence levels in their object detection or classification outcomes (as studied in heterogeneous opinion dynamics with uncertainty) [17]. Despite its growing practical importance, this form of uncertainty remains insufficiently addressed in decentralized robotic consensus research. In this work, uncertainty is defined as uncertainty in the exchanged class-probability vectors; in the evaluation (Section 5), we abstract sensing and model variability through stochastic confidence outputs and injected false detections, while a dedicated empirical study of heterogeneous model disagreement is left for future work.

From a practical perspective, these challenges are critical in applications such as search and rescue, hazardous material detection, infrastructure inspection, and planetary exploration, where reliable collective perception is essential for safe and effective operation. In such scenarios, premature or unstable consensus can lead to incorrect task allocation, missed hazards, or inefficient resource deployment.

The research gap addressed in this work lies in the absence of a decentralized consensus mechanism that (i) explicitly quantifies uncertainty in a mathematically grounded manner, (ii) adaptively modulates influence based on evidence strength, and (iii) operates without requiring predefined global robustness parameters. Existing approaches typically satisfy some of these properties but not all simultaneously. This motivates the development of a method that integrates uncertainty directly into the aggregation mechanism to enable robust, adaptive, and scalable consensus formation in heterogeneous autonomous robot teams.

#### ***1.4 Research Objective and Contributions***

Building on the research gap identified in the previous subsection, this work addresses the lack of decentralized consensus mechanisms that explicitly account for uncertainty in sensing, communication, and perception. The overall objective is to enable autonomous, heterogeneous robot teams to establish reliable shared context models under real-world conditions characterized by uncertain information.

To achieve this goal, the Decentralized Belief Consensus (DBC) algorithm is introduced. DBC integrates probabilistic reasoning with information-theoretic measures of uncertainty to improve robustness during collective decision-making.

The main contributions of this work are as follows:

1. Requirement-driven analysis: Derivation of a comprehensive set of functional and uncertainty-related requirements for decentralized context modeling in autonomous, heterogeneous robot teams, emphasizing robustness, scalability, and trust-aware information fusion.

2. Novel consensus method: Proposal of the DBC algorithm, which integrates entropy-based certainty measures into the consensus process to achieve deferred and adaptive agreement under uncertain conditions.
3. Simulation-based evaluation and benchmark comparison: Implementation of DBC in a simulation framework representing heterogeneous UAV–UGV teams and evaluation of its performance, robustness, and scalability under varying levels of sensor noise, communication delay, and false detections against established consensus algorithms. Both the simulation model and all implemented methods are openly available.

### 1.5 Paper Organization

The remainder of this paper is organized as follows. [Section 2](#) defines the decentralized context modeling problem and derives the functional and uncertainty-related requirements for autonomous, heterogeneous robot teams. [Section 3](#) reviews related work on consensus formation under uncertainty. [Section 4](#) introduces the proposed DBC algorithm. [Section 5](#) presents the experimental setup, simulation environment, evaluation metrics, and results, and discusses the findings and implications. [Section 6](#) concludes the paper and outlines directions for future research.

## 2 Problem Description and Requirements

An autonomous robot team can leverage a decentralized context model to perceive a collective understanding of their environment. In the field of collective perception, the team's objective is to detect, confirm, and interpret environmental objects distributed across a dynamic and partially unknown environment. Consider the scenario illustrated in [Fig. 1](#), in which a team of heterogeneous robots consisting of UAVs and UGVs explores a shared terrain in search of relevant environmental features such as obstacles [\[7\]](#), hazardous materials [\[2\]](#), or other mission-related objects. These objects of interest are associated with context attributes such as position, type, and detection confidence.

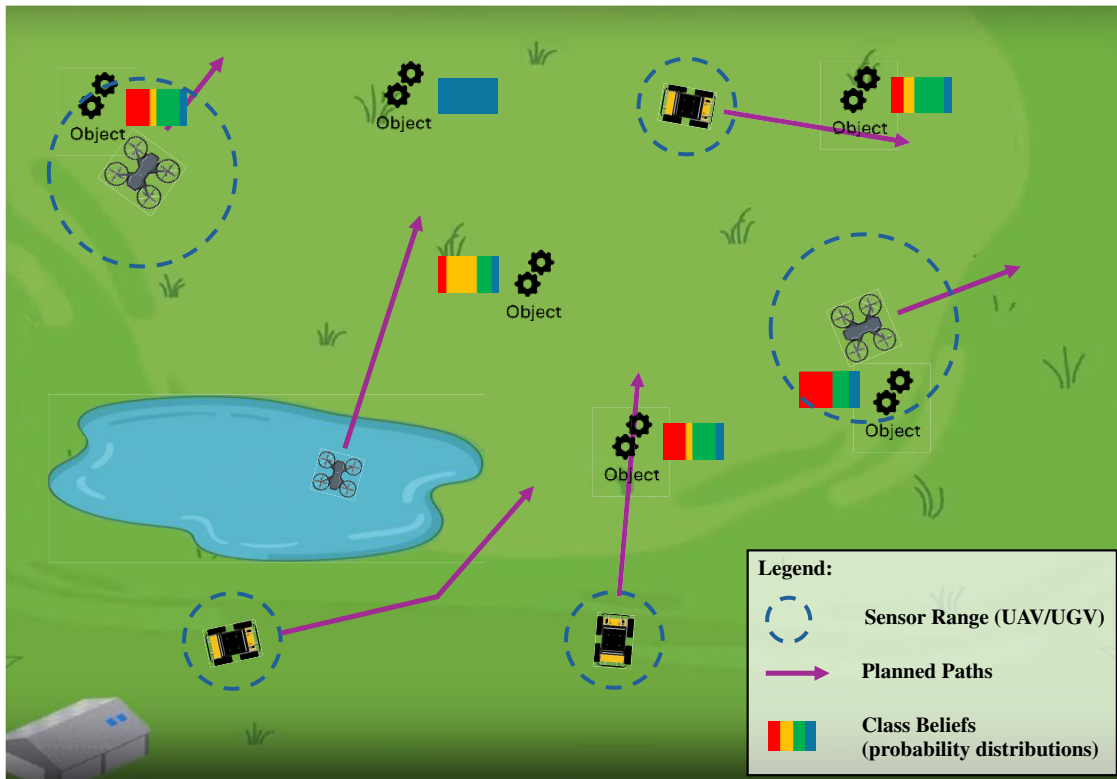
UAVs can be equipped with vision-based sensors and benefit from higher speeds and unobstructed flight paths, making them ideal for rapid, wide-area surveillance [\[12\]](#). However, they are limited in classification reliability and sensor precision due to altitude and motion-induced disturbances. In contrast, UGVs offer close-range sensing, higher observation accuracy, and verification capabilities, but are constrained by terrain and lower mobility [\[4\]](#).

Each robot navigates the environment autonomously, limited by its mobility, sensor range, and field of view. Environmental challenges include static obstacles (e.g., water bodies or buildings), occlusions, sensor blind spots, and dynamic phenomena such as moving or temporarily visible objects. As robots explore and observe their surroundings, they generate local, uncertain, and often incomplete observations. These observations are asynchronously shared within the team via decentralized communication with stochastic delays and possible packet loss as common in uncertain networks [\[13\]](#).

In the absence of a central coordinator or global map, the robots must collaboratively fuse their distributed beliefs into a consistent, shared world model. For such collective perception, decentralized consensus formation is necessary to robustly aggregate uncertain, asynchronous, and partially overlapping observations from multiple heterogeneous robots into a reliable decentralized context model. The following assumptions are made for the considered autonomous robot team:

- A1: The environment contains a finite set of objects, associated with spatial context (e.g., coordinates) and semantic properties.

- A2: There is no central control authority. All decisions are based on local and shared information within the robot team.
- A3: The robot team consists of UAVs and UGVs with different sensory, mobile, and functional characteristics.
- A4: Each robot has only limited and uncertain information about the environment.
- A5: The decentralized context model can iteratively be refined based on local sensing and peer-to-peer communication.
- A6: All robots are cooperative and trustworthy. Any deviations from expected behavior are assumed to be caused by hardware faults, not malicious intent.



**Figure 1:** Visual illustration of a robot team perceiving the environment and exploring environmental objects.

## 2.1 Mathematical Problem Formulation

In this section, the decentralized context modeling and consensus problem in heterogeneous multi-robot systems is formally defined. The set of robots is denoted by  $R = \{r_1, r_2, \dots, r_N\}$ , where each robot represents a UAV or UGV with distinct capabilities. The environment is assumed to contain a set of objects, represented by  $O = \{o_1, o_2, \dots, o_L\}$ , as shown in Fig. 1. Each object  $o_k$  can be classified by any robot  $r_i$ , yielding a probability vector  $P_{r_i}^{o_k} = \{p_1, p_2, \dots, p_M\}$ , where  $p_M \in [0, 1]$  and  $\sum_{m=1}^M p_M = 1$ . Such probabilistic outputs naturally arise from modern perception pipelines (e.g., CNN-based detectors) and are commonly fused in decentralized estimation frameworks [18,19]. The probabilities represent the robot's belief over a predefined set of object classes  $C = \{c_1, c_2, \dots, c_m\}$ . For example, if  $C = \{\text{fallen tree, sinkhole, rock debris, clear}\}$ , then  $P_{r_i}^{o_k} = (0.60, 0.25, 0.10, 0.05)$  indicates that robot  $r_i$  considers object  $o_k$  most likely a fallen tree. Consensus within this paper is defined as the autonomous robot team agreeing on a specific object classification based on the locally generated probability distribution  $P_{r_i}^{o_k}$ .

Decentralized consensus is applied to derive an estimate of the context model  $\phi_{r_i}(t)$  for each robot, which contains the different perceived objects. This estimate is denoted by  $\hat{\phi}_{r_i}(t)$  and is obtained by fusing local and neighboring models through a consensus update procedure:

$$\hat{\phi}_{r_i}(t) = \mathbf{ConsUpd}(\phi_{r_i}(t), \{\phi_{r_j}(t - \delta_j)\}_{k \in N_i}) \quad (1)$$

Here,  $N_i \subseteq R$  denotes the set of neighboring robots with which  $r_i$  can exchange data, and  $\delta_k$  captures possible communication delays. The consensus update function can be based on various approaches, which are further explored in [Section 3](#).

Based on the estimated consensus model  $\hat{\phi}_{r_i}(t)$ , a task allocation policy is selected for each robot, linking state uncertainty and multi-robot planning [20,21]. This policy, denoted by  $\pi_{r_i}$ , is chosen to maximize the expected task utility given the current estimated consensus:

$$\pi_{r_i} = \mathbf{argmax}_{\pi \in \Pi_i} \mathbb{E}[u(\pi | \hat{\phi}_{r_i}(t))] \quad (2)$$

where  $\Pi_i$  is the set of possible policies for robot  $r_i$ , and  $u(\cdot)$  is a utility function evaluating the expected performance of the plan  $\pi$  under the consensus model  $\hat{\phi}_{r_i}(t)$ . In situations where significant changes in the consensus model are detected, a replanning step is initiated. This is triggered when the following condition is satisfied:

$$|\hat{\phi}_{r_i}(t) - \hat{\phi}_{r_i}(t-1)| \geq \epsilon \quad (3)$$

with  $\epsilon$  defining a threshold for acceptable deviation in the context model consistent with event-triggered replanning in decentralized settings [21].

It is noted that task planning and policy optimization have been extensively studied by the research community and remain an active research area due to their inherent complexity and computational demands. Given the complexity and depth of this topic, a detailed exploration of planning algorithms is beyond the scope of this work. Instead, the focus of this paper lies in the challenge of collective perception and consensus under uncertainty within autonomous robot teams, which serves as an important prerequisite for reliable decentralized planning and coordination.

All variables and formal symbols used in this problem formulation are listed and defined in [Table 1](#). This model provides the foundation for the development of cooperative perception and decentralized consensus methods in heterogeneous multi-robot systems.

## 2.2 Implications and Requirements

Based on available requirements of former research and the identified challenges in the areas of decentralized context modeling, handling of uncertain information and consensus formation, a set of requirements with focus on robustness under uncertainty is derived. These requirements outline the essential capabilities a method must fulfill to support autonomous robot teams involving heterogeneous robots, uncertain information, and delayed communication.

**Table 1:** Overview of mathematical variables.

Variable	Description
$R = \{r_1, r_2, \dots, r_N\}$	Set of robots, indexed by $i \in \{1, \dots, N\}$
$R^{UGV}, R^{UAV} \subseteq R$	UGV and UAV subsets
$O = \{o_1, o_2, \dots, o_K\}$	Set of environmental objects, indexed by $k \in \{1, \dots, K\}$
$P_{r_i}^{o_k} = \{p_1, p_2, \dots, p_M\}$	Probability distribution of robot $r_i$ for object $o_k$ , indexed by $m \in \{1, \dots, M\}$
$T = \{t_1, t_2, \dots, t_T\}$	Set of discrete time steps, indexed by $t \in \{1, \dots, T\}$
$\phi_{r_i}(t)$	Local context model of robot $r_i$ at time $t$
$\hat{\phi}_{r_i}(t)$	Estimated consensus model of robot $r_i$ at time $t$
$\pi_{r_i}$	Task policy for robot $r_i$

### 2.2.1 General Requirements

#### R1: Scalability in Large Robot Teams

It must be ensured that the method scales with the number of participating robots. Scalable consensus behavior in large teams has been emphasized as a fundamental requirement for large robot teams [6,22,23]. Performance and convergence behavior should remain stable even in large-scale teams.

#### R2: Trust-Aware Decentralized Consensus Formation

The method should enable decentralized robots to estimate the trust of specific contextual and mission-relevant information (e.g., the presence of a hazard at a location), which has proven effective in enabling interpretable and robust consensus [24]. This requires mechanisms for fusing local and possibly contradictory information into a coherent and interpretable consensus model.

#### R3: Low Dependency on Predefined Global Parameters

The method should not depend on prior knowledge. This reduces the need for manual tuning and makes the method more applicable to real-world, unpredictable deployments [18].

#### R4: Communication Efficiency

The method should perform with low number of communications and avoid unnecessary message passing or synchronization. Prior work highlights communication efficiency and robustness against latency as critical factors for real-world applicability [25]. Therefore, the consensus process should function with delayed information sharing.

### 2.2.2 Uncertainty-Specific Requirements

#### R5: Deferred Consensus Under Uncertainty

The method should prevent full commitment to a consensus decision when the available evidence is weak, inconsistent or marginal [6]. It should allow robots to maintain intermediate or undecided states until a sufficient level of collective confidence emerges.

#### R6: Uncertainty Due to Faulty Robots and Incorrect Information

The method must maintain stability and acceptable accuracy even if some robots provide incorrect or outdated information [22]. Smaller amounts of such inputs should not destabilize the global model or lead to incorrect consensus.

### **R7: Uncertainty Due to Sensor Observations**

Uncertainty resulting from inconsistent or ambiguous sensor inputs must be explicitly handled. Local information are expected to fluctuate over time due to environmental or sensor-induced variation [12]; however, global consensus stability should be preserved.

### **R8: Uncertainty Due to Dynamic Group Composition**

The method must not rely on prior knowledge of the number or identity of all participating robots. It must tolerate partial participation, intermittent dropout, and dynamically changing group composition during runtime [25].

## **3 Related Work**

This section provides an overview of existing work on uncertainty handling and consensus formation in decentralized autonomous robot systems. Based on a structured analysis of relevant literature, a classification is developed that distinguishes between approaches without communication (indirect interaction) and those with explicit communication (direct interaction). Furthermore, relevant perspectives from adjacent disciplines are examined. Finally, the most promising methods are evaluated in terms of the requirements identified in [Section 2](#).

### **3.1 Review Methodology**

The literature review was guided by a structured and multi-layered methodology. As a starting point, several published survey articles on uncertainty handling [24,26], decentralized multi-robot systems [27], and consensus formation [6,28] were analyzed to define key concepts and terminology. Building upon these foundations, a systematic literature review was conducted using three major academic databases: IEEE Xplore, Web of Science, and Scopus.

The following search term was applied:

(“robot system” OR “robot swarm” OR “swarm robotics” OR “collective robotics”) AND (“decentralized” OR “distributed”) AND (“uncertainty” OR “stochastic” OR “noisy” OR “noise” OR “fault tolerance”)

AND (“consensus” OR “collective decision making” OR “collective estimation” OR “collective perception”)

Only peer-reviewed papers written in English and published within the last ten years were included. To broaden the perspective, similar searches were carried out in adjacent domains—specifically road-traffic systems and cyber-physical networks—without restricting the scope to autonomous robots. Additionally, a forward and backward search strategy was applied to capture relevant work that uses non-standard terminology. Full-text analysis was performed on the final selection, with a focus on approaches addressing uncertainty in multi-robot (multi-vehicle, multi-agent) decision-making through consensus mechanisms.

Two major categories of consensus formation were identified:

1. consensus without explicit communication ([Section 3.2](#)), typically relying on environmental or stigmergic interaction (i.e., insect behavior), and
2. consensus with explicit communication among robots ([Section 3.3](#)).

Additionally, consensus models from adjacent fields such as medicine, psychology, neuroscience, and economics were identified and discussed in [Section 3.4](#). Approaches based on blockchain were explicitly excluded, as their assumptions on adversarial, non-cooperative agents are incompatible with the cooperative nature of robot teams assumed in this work.

### 3.2 Consensus with Indirect Interaction

In swarm robotic systems, various forms of indirect interaction for consensus formation and collective decision-making have been demonstrated without requiring explicit inter-robot communication. These behaviors are typically inspired by biological systems and emerge from local interactions with the environment or physical coupling. The absence of communication reduces system complexity and lowers hardware requirements, making these strategies attractive for large-scale deployments with very limited communication resources.

One example that uses environmental stimuli such as light, temperature, or audio signals, is the BEECLUST algorithm [29], which mimics honeybee behavior to enable robots to aggregate in optimal zones. The authors extended this with a fuzzy-based controller, improving robustness against noise and allowing more precise directional responses through fuzzy logic mechanisms.

Physical interaction has been explored in [30] as a purely non-communicative mechanism for self-organization. Pan et al. demonstrated that simple mobile robots equipped with magnets could self-aggregate through magnetic attraction alone. These systems required no sensors or computing resources and were driven solely by mobility and physical forces.

Another study [31] has shown that collective perception and gradient following can be achieved via indirect interactions. For example, swarm drones were shown to be capable of performing collective gradient sensing by modulating their trajectories according to locally sampled values, without any shared memory or communication.

A variety of indirect consensus behaviors have been classified as “emergent” in [32]. Vega and Nowzari emphasized the observer-dependence of classifying such behavior as emergent or swarm-like. Their taxonomy distinguishes between nominal, weak, and strong emergence, highlighting that externally visible coordination may not imply true swarm intelligence without internal feedback mechanisms.

Finally, an automatic behavior design framework such as the one proposed by Salman et al. [33] has optimized stigmergy-based swarm behavior, leveraging evolutionary strategies to encode environment-mediated interactions without requiring direct communication.

Collectively, these approaches illustrate how consensus or coordination can emerge from indirect interactions, relying entirely on local perception, stochastic motion policies, and physical coupling—without the need for explicit communication between robots.

### 3.3 Consensus with Direct Interaction

In contrast to indirect interaction methods, approaches in this section assume that robots interact directly by communicating explicitly to share information. A key distinction is made between two subcategories: Collective Task Allocation and Collective Belief Generation.

Collective Task Allocation (Section 3.3.1) focuses on planning and coordinating robot tasks within the robot team. While these works are closely related to the problem considered in this paper, they are included primarily for completeness. Consequently, the reviewed approaches are only briefly discussed and evaluated. Moreover, due to the large body of existing research in this area, only a small and representative subset of recent contributions is considered.

Collective Belief Generation (Section 3.3.2), in contrast, directly addresses the core problem investigated in Section 2, namely the generation of a coherent shared representation of the environment, often formalized as decentralized context models. These works are therefore analyzed in greater detail.

### 3.3.1 Collective Task Allocation

A comprehensive theoretical foundation for robot planning and coordination under uncertainty is given in [20]. The book discusses probabilistic decision-making frameworks such as Partially Observable Markov Decision Processes (POMDP), decentralized Markov Decision Processes (MDP), and Bayesian games, which are increasingly used to model inter-robot coordination under uncertainty. A comprehensive treatment of multi-robot coordination methods under uncertainty—ranging from rule-based to market-driven approaches—is also provided in [21].

The work in [34] introduces a distributed coalition formation game for heterogeneous multi-robot systems. Robots form coalitions based on local beliefs about incomplete task information. A belief update mechanism allows robots to refine their understanding of task types over time, enabling convergence toward Nash-stable coalitions. This approach combines coalition game theory with distributed belief learning, offering robustness in surveillance applications where task definitions are not initially known.

A proactive task allocation method under spatiotemporal uncertainty is proposed in [35]. They model task announcement as a continuous-time Markov process and integrate it with an extended Sequential Single-Item (SSI) auction framework. Robots are proactively positioned at intermediate waiting points based on predicted task locations, significantly reducing service delays. This method demonstrates how learned probabilistic models can inform decentralized planning.

A hierarchical planning scheme is introduced by [14] in the form of the SCoBA algorithm, which separates individual policy generation from inter-robot conflict resolution. This method enables online reallocation of tasks and stochastic success rates. The two-layer architecture yields strong performance with theoretical guarantees and practical scalability.

Task coordination in dynamic mission scenarios is addressed by [36] using an enhanced Consensus-Based Bundle Algorithm (CBBA). This approach allows mission reallocation in UAV teams in response to robot failures. A heartbeat mechanism ensures timely detection of UAV loss, and a timeliness-aware bidding strategy is introduced to accelerate convergence. The method improves on classical CBBA by increasing robustness and responsiveness to real-world uncertainties. Similar to CBBA-based approaches, [37] introduces a robust task allocation mechanism incorporating probabilistic failure models into an auction-based framework. By modeling risk-aware bidding strategies, their system increases resilience against execution uncertainty without requiring centralized coordination.

Lastly, hierarchical task allocation under execution uncertainty is explored in [38]. A two-level hindsight optimization method is presented: the inner loop handles deterministic assignment, while the outer loop samples failure scenarios to proactively plan for contingencies. This enables the anticipation and mitigation of uncertainty without overburdening real-time computation.

These contributions demonstrate the diversity of techniques for consensus-based coordination in multi-robot systems. They address challenges such as scalability, resilience, and uncertainty mitigation through dynamic planning, learning, and communication.

### 3.3.2 Collective Belief Generation

The Linear Consensus Protocol (LCP) [25] represents a foundational mechanism for distributed agreement in multi-agent systems, particularly where agents iteratively adjust their states based on local neighbor information. Originally formulated to address average-consensus problems under both fixed and switching topologies, LCP ensures asymptotic convergence provided that the underlying network is strongly connected and, for average consensus, additionally balanced. The authors rigorously analyzed LCP under various conditions, showing its convergence rate depends on the algebraic connectivity (Fiedler eigenvalue)

of the network's mirror graph. While elegant and simple, LCP lacks robustness against adversarial behavior and struggles in dynamic environments with time delays unless topology and weights are carefully tuned.

The Weighted-Mean Subsequence Reduced (W-MSR) algorithm [39] is a fault-tolerant consensus protocol for coordinating agents despite faulty or malicious participants. Unlike standard averaging, W-MSR applies local filtering: at each step, an agent discards the  $F$  largest and  $F$  smallest neighbor values, then computes a weighted average of the remainder. This shields updates from manipulated or corrupted data and requires only local information—no knowledge of global topology or misbehaving identities. A central insight from [39] links correctness to a structural property called robustness. Under the  $F$ -total threat model (up to  $F$  adversaries anywhere), the network must satisfy that for every partition into two non-overlapping groups, at least one group contains a node with more than  $F$  incoming neighbors from outside the group. This prevents any subgroup from being fully insulated and dominating via coordinated misinformation. The condition is stricter than classical connectivity and fits decentralized, large-scale, partially observable settings. Despite its advantages, W-MSR has several important limitations. Filtering can remove legitimate but extreme measurements, reducing accuracy—especially in small networks or when outliers are informative. Convergence is typically slower than in non-resilient methods because part of the data is discarded each round. Success still hinges on nontrivial topology assumptions; sparse or weakly connected networks may fail the robustness requirement, particularly under dynamic or adversarial changes. Finally, W-MSR is non-adaptive: the parameter  $F$  is fixed a priori. If the true number of faulty nodes is lower, the algorithm discards too much information, causing unnecessary performance loss.

The approach for Collective Decision through Cross-Inhibition (CDCI) [22] is inspired by nest-site selection in honeybee swarms but differs from many nature-based swarm approaches by relying on explicit inter-robot communication. Each robot operates as a probabilistic finite state machine, switching between uncommitted and committed states. Key transitions—such as recruitment and cross-inhibition—require direct message exchanges to influence the commitment state of robots. This communication enables faster convergence, particularly when options are close in quality, and helps break decision deadlocks. This versatility makes CDCI applicable in diverse robotic scenarios. However, CDCI is not without limitations. First, its effectiveness hinges on accurate estimation of option quality and population ratios by individual robots, both of which are challenging in noisy or dynamic environments. Second, the strength of cross-inhibition must be carefully calibrated: if too weak, deadlocks may persist; if too strong, the system may become unstable or oscillate. Third, the model assumes well-mixed interaction patterns, which are rarely realistic in spatially constrained systems. Deviations from this assumption can lead to mismatches between predicted and actual outcomes. Lastly, while CDCI enables scalability, its probabilistic design can introduce stochastic variability that may compromise predictability in safety-critical applications.

The Distributed Bayesian Belief Sharing (DBBS) algorithm [18] is a probabilistic approach to multi-option consensus in decentralized robot teams. Instead of single-choice votes, DBBS maintains a full belief vector for each robot over all possible options and updates it over time using Bayesian statistics. Robots exchange compressed versions of these belief vectors with neighbors and integrate them via a weighted fusion rule, enabling the group to approximate a posterior distribution over the discrete state space without centralized control or raw-data sharing. DBBS's strengths include scalability and robustness in noisy or feature-skewed environments. By preserving distributional information, it reduces premature lock-in and maintains useful diversity, supporting consensus even with sparse communication. Performance degrades gracefully as communication range shrinks or ambiguity rises, typically reaching acceptable decisions without hard failures. The trade-offs are higher communication and computational costs—transmitting belief vectors scales with the number of options—and sensitivity to parameter tuning, notably the fusion weight

$\mu$  and memory decay  $\lambda$ . Poor settings can amplify feedback and drive inaccurate consensus. On resource-limited hardware, memory and bandwidth constraints may require further optimization. Overall, DBBS offers a powerful, generalizable framework for belief-driven collective decision-making under uncertainty.

The Ranked Voting Decision Algorithm (RVDA) [40] is a decentralized strategy for forming a shared preference order among multiple options. Inspired by the Borda count method [41], each robot maintains a local ranking of sites derived from uncertain observations. Robots communicate in pairs and run local elections: each converts its ranking into scores, exchanges lists, and re-ranks accordingly. Iterating these pairwise updates steers the team toward a consensus ranking. Compared with belief-fusion methods, RVDA is lightweight in memory and bandwidth because robots exchange ordered lists rather than full probability tables. Experiments show strong robustness in noisy environments and good scalability with team size. By preserving rank-based diversity early on, RVDA reduces premature lock-in and often achieves lower error under high sensor noise. Drawbacks remain. Two-agent elections introduce stochasticity, increasing scatter in final rankings. Convergence is typically slower than deterministic fusion, especially when uncertainty is low. Accuracy can also degrade when evidence rates are too sparse or too dense, which disrupts the balance between observation and opinion exchange. Despite these limits, RVDA is a practical, noise-resistant approach for collective decisions in uncertain, multi-option settings.

The Maximum Likelihood Estimate Sharing (MLEs) algorithm is a decentralized strategy for collective estimation in robot teams, focused on achieving accurate consensus with minimal communication overhead [42]. Each robot independently estimates environmental properties by modeling sensor readings as a Bernoulli process. These individual estimates are periodically shared with neighbors and fused into a collective estimate using a dynamic weighting function that reflects the confidence in each observation. Robots with many samples have more influence, while those with few remain cautious, allowing the team to adaptively reinforce high-confidence information. MLES offers a strong balance between speed and accuracy, even in uncertain or spatially challenging environments. Its robustness stems from its confidence-weighted self-organization, which suppresses premature consensus convergence and resists misinformation.

However, MLES has limitations. It is currently tailored for binary environments with only two features, lacks a mechanism for dynamic adaptation to multi-feature settings, and still depends on careful tuning of decay parameters. In larger or more heterogeneous teams, improperly tuned parameters may degrade performance. Despite these, MLES presents a lightweight and scalable framework well-suited for smaller robot systems operating under uncertainty.

The Expectation-based Extreme Value (EEV) algorithm [7] is a decentralized consensus method for robot teams operating amid obstacles and potentially malicious robots. EEV combines two mechanisms: a local expectation over neighbors' opinions and a thresholded "extreme value" rule. Each robot computes an expected direction (e.g., left vs. right) from its own and neighbors' views; if the expectation is clearly positive or negative, it commits fully to that direction, otherwise it maintains its current choice. EEV is particularly effective against Byzantine behavior. Rather than fixed trimming or sorting, it adapts by re-evaluating expectations each step. Experiments with Byzantine robots show EEV reaching valid consensus where LCP and W-MSR often converge to invalid outcomes. It can also distinguish valid from invalid Byzantine robots without prior knowledge of their number or identity. Limitations include dependence on sufficient connectivity and reliable local perception. The approach assumes binary decisions, limiting direct extension to multi-option settings. And while less parameter-sensitive than W-MSR, EEV still embeds implicit thresholds that may need empirical tuning. Overall, EEV is robust, adaptive, and computationally simple for consensus under uncertainty.

Taken together, the reviewed algorithms illustrate the diverse strategies and trade-offs involved in achieving decentralized consensus for belief generation, ranging from simplicity and speed to robustness and probabilistic accuracy.

### 3.4 Requirements Alignment and Scientific Gap

To evaluate the suitability of existing consensus algorithms for decentralized context modeling under uncertainty, a structured requirements-based assessment was conducted. Table 2 summarizes this alignment across the eight requirements (R1–R8), as defined in Section 2.2. It is important to emphasize that the requirements alignment exclusively covers methods exploiting communication between the robots within the category of Collective Belief Generation.

**Table 2:** Alignment of requirements. Fulfillment grades: ++ (excellent), + (good), o (moderate), - (weak) and – (poor).

Method	Source	Year	R1	R2	R3	R4	R5	R6	R7	R8
LCP	[25]	2004	+	o	+	++	o	–	–	–
W-MSR	[39]	2013	o	o	–	+	–	+	o	–
CDCI	[22]	2015	o	+	–	o	+	–	+	o
DBBS	[18]	2021	–	+	–	–	+	o	+	+
RVDA	[40]	2023	+	–	–	–	–	–	+	o
MLES	[42]	2023	++	+	o	+	+	–	–	o
EEV	[7]	2024	+	++	+	++	–	+	–	o

The analysis reveals a consistent gap that none of the evaluated methods fulfill all requirements to a sufficient degree under realistic robotic conditions. While individual algorithms demonstrate partial strengths—such as communication efficiency (e.g., LCP, EEV) or robustness against faulty robots (e.g., W-MSR, EEV)—they often rely on restrictive assumptions, predefined global parameters, or lack mechanisms for deferred consensus under uncertainty. Furthermore, deferred consensus (R5), dynamic participation handling (R8), and minimal dependency on prior knowledge (R3) are rarely addressed in combination.

This misalignment highlights the lack of robustness and generalizability in existing methods, which are required for realistic decentralized consensus for autonomous robots. Thus, these findings motivate the development of a new approach that explicitly addresses the full set of requirements and operates reliably in uncertain, dynamic multi-robot environments without centralized control.

### 3.5 Entropy-Based Adaptive Consensus under Uncertainty

The analysis presented before clearly shows that existing methods of consensus formation in decentralized robot systems do not adequately meet the requirements. In particular, there is a lack of adaptive mechanisms for dealing with uncertainty when evidence is limited. To address this gap, this section focuses specifically on information theory as a formal framework for modeling uncertainty.

Information theory is the mathematical study of the quantification, storage, and communication of information. The field was established and formalized by Claude Shannon in the 1940s, who also introduced the concept of entropy in his foundational work in 1948 [43]. Entropy quantifies the uncertainty in a message source and measures the expected “surprise” of an information [44]. The Entropy for an object  $o_k$  of robot

$r_i$  can be expressed as

$$H_{r_i}^{o_k} = - \sum_{m=1}^M p_m \cdot \log_2(p_m) \quad (4)$$

In (4),  $p_m$  denotes the probability assigned to the object  $o_k$  for  $M$  different options within the probability distribution.

Entropy has been used to evaluate consensus strength and decision confidence in various fields, i.e., informatics [45], economics [46,47], and autonomous road traffic [48,49]. These studies use entropy to measure dissent, guide aggregation of diverging opinions, and evaluate convergence reliability. In robotic systems, entropy-based metrics can quantify confidence levels in collective decisions, enabling robots to delay or reinforce commitment depending on their local certainty. Due to its mathematical precision and adaptability to probabilistic environments, entropy-based approaches are well-suited for robotic consensus under uncertainty.

### 3.6 Summary of Findings

This section provided a structured analysis of existing consensus methods in decentralized robot systems, highlighting a clear gap between current capabilities and the requirements for robust consensus formation under uncertainty. Despite diverse algorithmic strategies, no method fully addresses all requirements, particularly allowing deferred consensus, dynamic group composition, and uncertainty-aware belief commitment. To address this limitation, the analysis turned specifically to information theory as a formal framework for uncertainty quantification. Information-theoretic concepts, and in particular Shannon entropy, offer a mathematically rigorous measure of belief evaluation under uncertainty. Entropy enables robots to assess the confidence of their own and others' observations, supporting more informed, deferred consensus decisions. By integrating entropy into the decision process, robots can dynamically modulate their behavior based on evidence strength, avoiding premature convergence. This insight forms a foundational element for the decentralized belief consensus method proposed in the following chapter.

## 4 Proposed Approach

This section introduces a Decentralized Belief Consensus (DBC) algorithm based on related work findings to address the problem described in Section 2. First, an overview of the method is given. Then, a more detailed explanation of the incorporated steps is explained.

### 4.1 Method Overview

The method, as depicted in Fig. 2, operates as an asynchronous, event-driven process in which each robot continuously updates its information about the operational environment. Local observations and received messages are handled in parallel and independently trigger belief updates (Belief Generation). Once new information is available, the robot performs a local update and shares its updated belief with neighboring agents during Belief Distribution. In the Belief Balancing step, incoming beliefs are integrated using confidence-weighted fusion to reduce conflicts and incorporate multi-robot perspectives. Subsequently, the robot aggregates all collected beliefs into a consistent internal estimate (Belief Aggregation). Finally, an optional Belief Optimization step refines this estimate to produce an updated representation of the environment.

The certainty value based on entropy not only serves as an additional reliability (certainty) metric, but also directly influences consensus dynamics. Shannon entropy quantifies the dispersion of a robot's

belief distribution. High entropy corresponds to a flat probability distribution, indicating high predictive and potentially epistemic uncertainty, while low entropy reflects a peaked distribution and strong belief conviction. At the system level, this mechanism induces adaptive convergence behavior: Robots with more concentrated beliefs (lower entropy) receive higher relative influence in the fusion step, while high-entropy beliefs contribute less. This prevents weak or ambiguous evidence from dominating the aggregated belief and supports deferred commitment under uncertainty. As evidence increases and entropy decreases, certainty increases and the fusion progressively emphasizes reliable beliefs, thereby accelerating convergence toward consensus. Thus, entropy acts as a self-regulating mechanism that dynamically balances stability and determination in the consensus process.

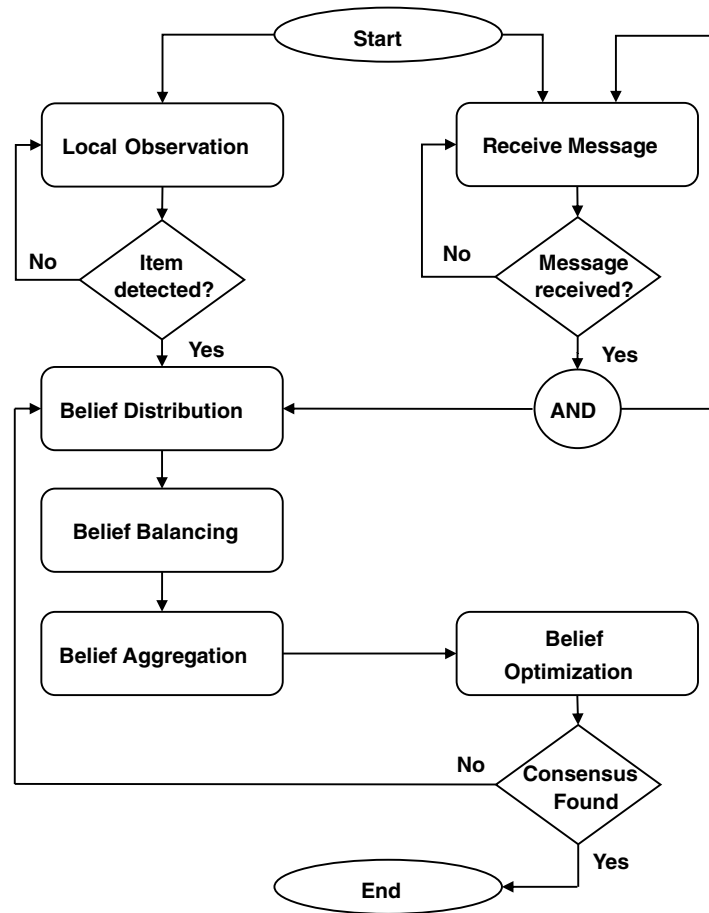


Figure 2: Flowchart of decentralized belief consensus (DBC).

#### 4.2 Method Details

Each robot maintains a probabilistic belief distribution  $P_{r_i}^{o_k}$  over a finite set of object classes. To quantify the uncertainty of this belief, Shannon entropy is computed according to Eq. (4). The entropy is mapped to a bounded certainty value

$$\Omega_{r_i}^{o_k} = \frac{1}{1 + H_{r_i}^{o_k}} \tag{5}$$

which yields

$$\frac{1}{1 + \log_2 M} \leq \Omega \leq 1 \quad (6)$$

where  $M$  denotes the number of classes. Lower entropy results in higher certainty, allowing each robot to assess the reliability of its current belief. We choose the monotone mapping  $\Omega = 1/(1 + H)$  because it is bounded and parameter-free; alternative normalizations are possible.

Belief distributions and their associated certainty values are exchanged asynchronously within the local neighborhood  $N_i$ . For readability, the object index  $o_k$  is omitted in the following equations. Upon receiving neighbor information, robot  $r_i$  computes certainty-weighted fusion coefficients:

$$\omega_i = \frac{\Omega_i}{\Omega_i + \sum_{j \in N_i} \Omega_j} \quad (7)$$

$$\omega_j = \frac{\Omega_j}{\Omega_i + \sum_{k \in N_i} \Omega_k}, \forall j \in N_i \quad (8)$$

The updated belief distribution is obtained by:

$$p_i^{\text{new}} = \omega_i \cdot p_i + \sum_{j \in N_i} \omega_j \cdot p_j \quad (9)$$

This formulation guarantees several desirable properties. First, the updated belief remains a valid probability distribution, as the weighted combination of normalized values preserves normalization. Second, robots with higher certainty contribute more strongly to the aggregated result, ensuring that well-supported information dominates the consensus process. Third, uncertain or inconsistent inputs have only limited influence due to their low certainty weights, which enhances robustness against ambiguous or unreliable observations. Finally, the method is fully decentralized and supports asynchronous execution, making it well suited for dynamic environments where robots operate independently. If a discrete class label is required, it can be obtained by  $\text{argmax}$  over  $p_i^{\text{new}}$ ; optionally, a confidence threshold on  $\Omega$  can be used to postpone commitment under weak evidence.

Optionally, a belief optimization step incorporates a certainty-weighted collective expectation, mapped to a binary extremum value. A small fraction of this extremal signal is mixed with the probabilistic belief via convex combination. That is, as a concrete hybrid extension, one may set  $m^* = \text{argmax}(p_i^{\text{new}})$ , where  $m^*$  is the class index of the object class with the highest probability, form the one-hot vector  $e_{m^*}$ , and mix it via  $p_i^{\text{opt}} = (1 - \alpha) \cdot p_i^{\text{new}} + \alpha \cdot e_{m^*}$ , with  $\alpha \in [0, 1]$ . This hybrid extension accelerates convergence while retaining uncertainty-aware fusion.

### 4.3 Computational and Communication Complexity

The computational and communication complexity of DBC depends on three key quantities: the number of robots  $N$ , the local neighborhood size  $|N_i|$ , and the number of object classes  $M$ .

During one update step, four main computational operations are performed. First, the Shannon entropy  $H_i^{(o_k)}$  is computed over the  $M$  class probabilities. This requires a linear pass over the belief vector and therefore scales proportionally to  $M$ . Second, the entropy is transformed into a scalar certainty value  $\Omega_i^{(o_k)}$ . This mapping is a constant-time operation. Third, robot  $r_i$  aggregates the certainty values of its neighbors  $r_j \in N_i$ . The normalization term used in the weights depends on the sum of all  $\Omega_j$  within the local neighborhood.

This step scales linearly with the number of neighbors  $|N_i|$ . Fourth, the belief aggregation step combines the local belief  $p_i$  with the neighbors' beliefs  $p_j$ . For each neighbor, a weighted combination over all  $M$  class values is performed. This is the dominant cost and scales proportionally to  $|N_i| \cdot M$ . Therefore, the per-object update complexity for robot  $r_i$  scales overall as  $\mathcal{O}(|N_i| \cdot M)$ . If a robot maintains beliefs over multiple objects  $|O_i|$ , the total cost grows linearly with  $|O_i|$ . In an event-driven implementation, such an update is triggered by either (i) a new local observation or (ii) reception of a neighbor message affecting the corresponding object.

Importantly, DBC depends on the local neighborhood size  $|N_i|$ , not directly on the total number of robots  $N$ . In typical decentralized robot teams with local communication,  $|N_i|$  remains bounded even if  $N$  increases. Under this common assumption of sparse connectivity, the per-robot computational effort scales approximately linearly with  $M$  and the number of locally maintained objects.

Regarding communication, each robot transmits for every object the belief vector  $p_i^{(o_k)}$  of size  $M$  and the scalar certainty  $\Omega_i^{(o_k)}$ . Thus, message size grows linearly with  $M$ . The communication load per robot scales with  $|N_i| \cdot M$ , while the total network load scales with the number of communication links.

In summary, DBC exhibits linear scaling in both computation and communication with respect to belief dimension and neighborhood size, supporting scalability in large but sparsely connected robot teams.

#### 4.4 Key Contributions and Requirements Alignment

The central contribution of DBC is the integration of Shannon entropy as a quantitative uncertainty measure directly into the consensus process. By transforming entropy into a bounded certainty value  $\Omega_i$ , belief influence is adaptively modulated during aggregation. Unlike classical linear averaging (e.g., LCP) or threshold-based extremum rules (e.g., EEV), DBC introduces a continuous, information-theoretic mechanism that attenuates uncertain inputs without discarding them. This directly addresses R5 (deferred consensus) and R7 (sensor uncertainty). In contrast to methods such as W-MSR, which require predefined bounds on faulty agents, DBC does not rely on global robustness parameters. Influence is derived solely from locally computed certainty values, enabling operation without prior knowledge of network size or adversary count. This supports R3 (low dependency on predefined global parameters) and R8 (dynamic group composition).

DBC maintains full probability distributions rather than binary states or scalar opinions. This preserves distributional information and reduces premature commitment. The convex certainty-weighted aggregation ensures that well-supported beliefs dominate while ambiguous or conflicting observations are naturally damped. This mechanism contributes to R2 (trust-aware consensus formation) and improves robustness to uncertain or low-confidence incorrect information (R6); systematic high-confidence biases remain a limitation.

The algorithm operates exclusively on local neighborhoods  $N_i$  and does not require centralized coordination. Computational and communication complexity scale with  $|N_i|$  rather than the total number of robots  $N$ , supporting R1 (scalability in large robot teams) and R4 (communication efficiency).

Taken together, DBC is not merely a combination of existing consensus strategies but introduces a structurally adaptive consensus paradigm grounded in information theory. By embedding uncertainty quantification directly into the aggregation mechanism, the method systematically bridges the gap identified in [Section 3.4](#) and provides a coherent, requirement-driven solution for decentralized context modeling under uncertainty.

## 5 Evaluation and Discussion

This section presents the simulation-based evaluation of DBC. The setup, simulation environment, evaluation metrics, and results are described to assess performance, robustness, and scalability under uncertain information.

### 5.1 Experimental Setup

The experimental evaluation is conducted through a simulation framework specifically developed for decentralized heterogeneous robot teams. The Python code used for the simulation, along with a brief execution guide, is available online ([https://github.com/hsu-aut/RIVA\\_Decentralized\\_Consensus](https://github.com/hsu-aut/RIVA_Decentralized_Consensus)). The primary objective is to assess the performance, robustness, and scalability of consensus methods under realistic environmental and operational conditions.

The mission of the robot team is to detect and localize 28 mines within an unknown terrain. Due to individual sensor limitations, occlusions, and environmental noise, no single robot can reliably confirm mine positions on its own. Therefore, a distributed consensus process is required to merge local observations into consistent shared beliefs.

The robotic team consists of a mix of UAVs and UGVs. In the standard configuration, each simulation includes one UAV and two UGVs, reflecting a typical reconnaissance-verification workflow in heterogeneous teams: UAVs provide rapid wide-area screening, while UGVs perform closer inspection and confirmation. This division of roles models practical deployments in which aerial platforms detect candidate targets and ground units verify them. To examine scalability, additional experiments are conducted with larger team sizes (see [Section 5.4.3](#)).

The perception model assumes vision-based object detection (e.g., CNN-based classifiers) producing probabilistic confidence values. In the presented mine-detection experiments, the classification is binary (mine vs. no-mine), i.e.,  $M = 2$ . To simulate sensor uncertainty in a controlled and reproducible manner, confidence scores are sampled from Gaussian distributions. The selected parameters (UAV:  $\mu = 0.75$ ,  $\sigma = 0.02$ ; UGV:  $\mu = 0.95$ ,  $\sigma = 0.02$ ) reflect the typical difference between high-altitude, motion-affected reconnaissance sensing and close-range ground verification. Gaussian sampling is used as an abstraction of softmax-based confidence outputs, where uncertainty manifests as dispersion around a mean confidence level. Samples are clipped to the interval  $[0, 1]$  to ensure valid probabilities, and the resulting belief vector is  $p = (p_{\text{mine}}, 1 - p_{\text{mine}})$ . While real sensor noise may not be strictly Gaussian, this model enables systematic variation of uncertainty while preserving comparability across methods.

To evaluate robustness against perception errors, false positives are intentionally injected into the sensor stream. The False Discovery Rate (FDR) represents the proportion of incorrect detections relative to true objects. Lower FDR levels correspond to moderate perception noise, while higher levels simulate challenging conditions such as occlusion, cluttered environments, lighting disturbances, or model misclassification. The upper range serves as a stress-test scenario to evaluate algorithmic stability under severe uncertainty.

All simulations run for 3000 s (50 min), approximating a mid-duration field mission. Each configuration is repeated 10 times to account for stochastic motion, perception variability, and asynchronous communication. Performance is evaluated using the metrics defined in [Section 5.3](#). In the event-driven implementation, belief updates are triggered by new local observations or by reception of neighbor messages affecting the corresponding object belief.

Three distinct experimental series are conducted:

1. Performance Evaluation (Section 5.4.1): Various consensus algorithms from Section 3.3.2—including LCP, WMSR, CDCI, DBBS, MLES and EEV—are compared against the proposed method DBC with respect to their convergence behavior and accuracy.
2. Robustness Analysis (Section 5.4.2): The impact of uncertain and misleading observations on consensus formation is evaluated by varying the False Discovery Rate (FDR), i.e., increasing the ratio of false positives.
3. Scalability Study (Section 5.4.3): The behavior of the consensus algorithms is analyzed under increasing team sizes to assess communication and convergence dynamics in large-scale deployments.

This structured experimental setup enables a comprehensive and reproducible comparison of consensus strategies under varying environmental complexity, uncertainty, and team composition.

## 5.2 Simulation Environment

The simulation environment models a 1600 m × 1000 m two-dimensional terrain that represents an unknown operational area containing buried mines, static obstacles, and varying terrain features. The environment is shared by a heterogeneous team of autonomous mobile robots, including UAVs and UGVs, which explore the area and collaboratively detect and verify mine positions.

All robots are free to navigate within the simulation area and operate independently according to their predefined search strategies. The UAVs execute a continuous U-shaped sweep pattern, enabling rapid and systematic area coverage. Due to its altitude and speed, it cannot interact with mines directly but contributes valuable initial observations from a broad vantage point. The UGVs perform an initial random exploration of the map. Upon receiving mine position information—either from their own observations or via communication—they switch to goal-directed navigation to verify reported mine locations. UGVs plan their paths using A\* path planning to navigate efficiently to target positions, while allowing them to avoid known static obstacles and mines.

The detection of mines is simulated based on probabilistic observations, which reflect the expected output of modern AI-based object detection systems [19,50]. Each observation results in a confidence value, which serves as input for the consensus formation process. These confidence values are sampled from normal distributions that differ by robot type. These distributions model both the reduced certainty of high-altitude UAV observations and the higher precision of low-altitude UGV sensing. Additionally, the simulation explicitly injects false positive detections (Section 5.4.2) to model perception errors and test the robustness of the decision-making process under uncertain information. The UAVs' and UGVs' motion speeds, sensor ranges, and perception capabilities differ and are summarized in Table 3. These differences reflect their distinct sensing roles within the heterogeneous team and are critical to the distributed fusion of information.

**Table 3:** Operational parameters.

Parameter	UAVs	UGVs
Velocity $v$ [km/h]	14.4	3.6
Detection range $r$ [m]	30	10
Perception Confidence	N (0.75, 0.02)	N (0.95, 0.02)

Inter-robot communication is modeled via an MQTT-inspired communication architecture with 4G network characteristics, including asynchronous message transmission and latency times of 50–100 ms. Robots exchange only relevant information, including newly detected mine positions and updates to their internal beliefs regarding known mines. No global synchronization or central coordination is assumed. This asynchronous, delay-prone communication model captures the practical challenges of distributed information exchange in real-world field deployments and serves as a basis for evaluating the robustness of the consensus methods.

### 5.3 Evaluation Metrics

To evaluate the performance of the consensus strategies under consideration, a set of metrics that jointly reflect the quality, accuracy, speed, and reliability of the decision-making process is defined. These metrics are partially adapted from the work proposed in [23] and are tailored to the scenario addressed in Section 5.1. In contrast to a global scalar consensus value, as it has been considered in previous publications [7,18,22,25,39,40,42], consensus is evaluated per object (e.g., mine) and the results are aggregated accordingly.

#### 5.3.1 Scatter during Consensus

The Scatter During Consensus quantifies the level of agreement among robots at a given time  $t$ . It is defined as the sum of squared Euclidean distances between the mean belief  $\bar{\phi}(t)$  across all robots (e.g., the existence of an object) and each robot's current belief  $\phi_{r_i}$  at that time:

$$\mathbf{ConsScatter}(t) = \sum_{i=1}^N [\bar{\phi}(t) - \phi_{r_i}(t)]^2 \quad (10)$$

This metric captures how dispersed or aligned the robot team is in its opinions. A lower scatter indicates a higher degree of internal agreement. The final ConsScatter refers to the internal agreement at the end of an experimental run.

#### 5.3.2 Error during Consensus

While the Scatter During Consensus measures internal agreement, the Error During Consensus evaluates how close the collective decision is to the ground truth. It is defined as the total absolute error between each robot's belief  $\phi_{r_i}$  and the known true value  $\phi_{true}$ :

$$\mathbf{ConsError} = \sum_{i=1}^N |\phi_{true} - \phi_{r_i}| \quad (11)$$

This metric provides a direct measure of the accuracy with respect to the correct environmental classification.

#### 5.3.3 Consensus Time

The Consensus Time measures how quickly the team reaches a sufficiently coherent collective decision. Rather than waiting for the absolute minimum of the scatter value—which may occur late or fluctuate due to noise—consensus time is the first time step  $t$  where the scatter drops below 5% of the initial scatter.

$$\mathbf{ConsTime} = \min_{t \in \mathbf{N}} \{t | \mathbf{Scatter}(t) \leq 0.05 \cdot \max_{t' \in \mathbf{N}} [\mathbf{Scatter}(t')]\} \quad (12)$$

### 5.3.4 Final Consensus Rate

In addition to timing and accuracy, the Consensus Rate quantifies the proportion of objects in the environment for which the robot team successfully reaches consensus. It is defined as the proportion of objects for which a valid consensus state is achieved according to the predefined consensus criterion. In Eq. (13), this metric is computed using an indicator function  $\mathbb{I}$ , which takes the value 1 if a valid consensus is achieved for a given object and 0 otherwise. The final consensus rate is the percentage of objects for which consensus was found by the end of the simulation. It provides a robust, object-level indicator of the algorithm's ability to generate consistent decisions across multiple targets and is particularly relevant in scenarios with a high number of distributed, independent perception objects.

$$\text{ConsRate} = \frac{1}{K} \sum_{k=1}^K \mathbb{I}(o_k) \quad (13)$$

## 5.4 Evaluation Results

In this section, the evaluation results are presented. The evaluation is structured into three parts: performance under ideal conditions, robustness against uncertain information and false detections, and scalability with respect to team size.

### 5.4.1 Performance Evaluation

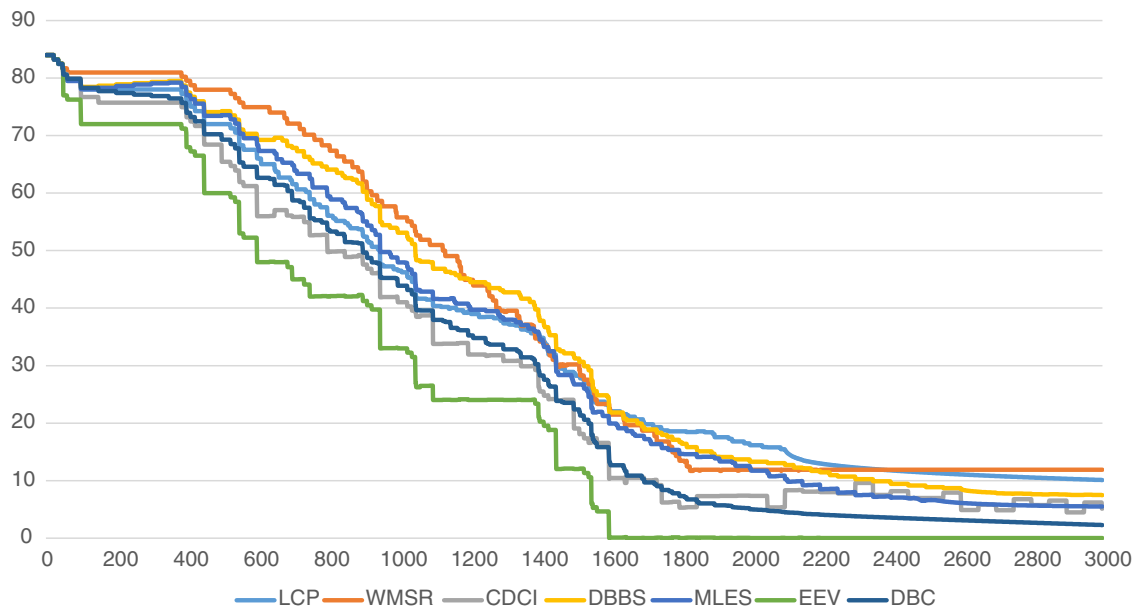
The first experiment investigates how effectively different consensus algorithms reduce uncertainty and reach agreement in the absence of artificial disturbances. The following methods from Section 3.3.2 were compared: LCP, WMSR, CDCI, DBBS, MLES, EEV, and the proposed method DBC.

The summarized performance metrics are shown in Table 4. These include the average consensus time, average final scatter, final consensus rate, and final error. DBC achieves a competitive consensus time of 63.6 s, markedly faster than WMSR (493.3 s) and slightly behind EEV (29.6 s), which benefits from its strong bias toward early decision-making. In terms of scatter, DBC reaches one of the lowest average values (0.004), indicating strong internal agreement within the robot team. All methods except CDCI achieve a 100% consensus rate in this ideal setup, and only EEV and DBC manage to reduce the final error below 3.

**Table 4:** Results of simulation experiments with regards to performance.

	LCP	WMSR	CDCI	DBBS	MLES	EEV	DBC
Avg. ConsTime [s]	82.4	493.3	80.1	79.7	83.3	29.6	63.6
Avg. ConsScatter	0.004	0.028	0.056	0.001	0.003	0	0.004
Final ConsRate	100%	100%	96%	100%	100%	100%	100%
Final Error	10.1	11.9	5.1	7.5	5.5	0	2.3

Fig. 3 illustrates the evolution of the total error over time for each method. All methods begin with similar initial error levels (which results from the 28 mines initially unknown to all robots) but differ in their consensus speed and final accuracy. The DBC approach demonstrates a steep error reduction early in the simulation and converges to a final error of 2.3, only surpassed by EEV, which achieves perfect accuracy (0.0) in this noise-free scenario. This perfect accuracy is achieved by the behavior of EEV, which jumps directly to the extreme perception value when a threshold value is exceeded. In contrast, traditional methods such as WMSR and LCP show significantly slower convergence and higher residual error levels.



**Figure 3:** Development of consensus error over time for various consensus methods.

These results confirm that DBC offers a competitive time with markedly lower error than most baselines; EEV is fastest but more extreme.

#### 5.4.2 Robustness Analysis

Controlled amounts of false positive detections were introduced into the simulation to assess the robustness of the tested consensus algorithms under uncertain information. The False Discovery Rate (FDR) defines the proportion of randomly inserted, incorrect mine detections and was varied in four levels: 0%, 16%, 33%, and 50%. This setup emulates false positive perception as it might occur with imperfect object detection systems in the field.

In this experiment, the analysis focuses on a representative subset of the previously tested consensus algorithms: LCP, CDCI, EEV, and the proposed DBC method. This selection reflects a diverse spectrum of algorithmic behavior observed in the performance evaluation. LCP and CDCI serve as well-established baselines with contrasting convergence strategies, while EEV and DBC represent the most promising methods in terms of accuracy and convergence time. The remaining approaches showed similar or inferior performance trends in earlier experiments and are therefore omitted here for clarity.

The results, shown in [Table 5](#), show clear differences in robustness. Results degrade substantially with rising FDR: for CDCI, the final consensus error increases from 5.1 (0% FDR) to 79.3 (50% FDR), while its consensus rate drops from 96% to 89%. LCP follows a similar but slightly less severe trend. By contrast, DBC's final error remains low and stable, rising only marginally from 2.3 to 2.6 across the full FDR range. Importantly, EEV does not maintain zero error at higher noise levels. While fast and decisive, EEV's final error increases from 0.0 at 0% FDR to 18 at 16%, 30 at 33%, and 84 at 50% FDR, reflecting its extreme-value commitment strategy under false positives. Thus, EEV is highly efficient but notably sensitive to uncertain information.

**Table 5:** Results of simulation experiments with regards to robustness.

	0% FDR				16% FDR				33% FDR				50% FDR			
	LCP	CDCI	EEV	DBC	LCP	CDCI	EEV	DBC	LCP	CDCI	EEV	DBC	LCP	CDCI	EEV	DBC
Avg. ConsTime	82.4	80.1	29.6	63.6	83.1	83.8	29.5	70.3	81.3	87.6	29.7	87.8	83.5	90.3	29.5	107.9
Avg. Cons Scatter	0.004	0.052	0	0.004	0.005	0.055	0	0.003	0.006	0.056	0	0.002	0.006	0.075	0	0.003
Final ConsRate	100%	96%	100%	100%	100%	93%	100%	100%	100%	90%	100%	100%	100%	89%	100%	100%
Final Error	10.1	5.1	0	2.3	20.5	25.4	18	2.4	16.8	47.2	30	2.5	18.8	79.3	84	2.6

In terms of reliability, both EEV and DBC sustain a 100% consensus rate under all tested FDR levels, consistently reaching agreement across the team despite injected noise. EEV is the fastest method in every condition, with consensus times stable around 29.5 s. DBC's consensus time increases from 63.6 to 107.9 s as FDR grows, yet still converges considerably faster than CDCI or LCP at higher noise levels. Final scatter values remain low and stable for both EEV and DBC, indicating strong internal alignment even under substantial uncertainty levels.

Overall, the results demonstrate that DBC achieves robustness comparable to that of EEV while offering better control over error dynamics and convergence behavior than traditional consensus mechanisms. The use of confidence-weighted belief fusion enables DBC to selectively attenuate the influence of unreliable or misleading observations, making it particularly well suited for deployment in uncertain and error-prone environments.

#### 5.4.3 Scalability Study

To analyze the scalability of the consensus algorithms, additional experiments were conducted with increasing team sizes, ranging from 3 to 12 robots. In each configuration, the ratio of UAVs to UGVs was kept consistent at 1:2, resulting in setups with 1 UAV + 2 UGVs, 2 UAVs + 4 UGVs, 3 UAVs + 6 UGVs, and 4 UAVs + 8 UGVs. In all scalability experiments, the FDR was fixed at 33% to ensure a realistic and challenging perception scenario for consensus formation.

The results in [Table 6](#) indicate that the increase of team size strongly impacts traditional consensus algorithms such as LCP and CDCI. As the team size increases, CDCI's average consensus time grows from 87.6 s with 3 robots to 232.8 s with 12 robots, accompanied by a significant rise in final consensus error, which reaches 104.1 in the largest configuration. Likewise, LCP also shows clear signs of degradation, particularly in terms of consensus accuracy and final error, which climbs to 133.5 with 12 robots.

**Table 6:** Results of simulation experiments with regards to scalability.

	3 Robots				6 Robots				9 Robots				12 Robots			
	LCP	CDCI	EEV	DBC	LCP	CDCI	EEV	DBC	LCP	CDCI	EEV	DBC	LCP	CDCI	EEV	DBC
Avg. ConsTime	81.3	87.6	29.7	87.8	83.1	131.1	30.4	90.6	85.6	225	30.5	92.1	88.3	232.8	32.7	105.4
Avg. Cons Scatter	0.006	0.056	0	0.002	0.013	0.104	0	0.001	0.012	0.269	0	0.003	0.013	0.314	0	0.005
Final ConsRate	100%	90%	100%	100%	96%	75%	100%	100%	96%	61%	100%	100%	93%	68%	100%	100%
Final Error	16.8	47.2	30	2.5	36.3	50.8	42	5.6	84.9	80.9	72	11.2	133.5	104.1	96	15.8

In contrast, the EEV and DBC methods maintain consistent and robust performance as team size increases. EEV continues to produce the fastest convergence times—remaining below 33 s across all setups—and achieves a final consensus rate of 100% in all cases. However, EEV's final error does increase noticeably with larger teams, reaching 96.0 with 12 robots, suggesting that while it is fast and reliable in forming agreement, its accuracy suffers in larger teams under uncertain conditions.

DBC again demonstrates the best balance of speed, precision, and reliability. Although its average consensus time rises from 87.8 to 105.4 s as the team scales from 3 to 12 robots, the increase remains moderate compared to CDCI or LCP. Moreover, DBC consistently maintains low final scatter values, a 100% consensus rate, and a final error of just 15.8 in the 12-robot scenario. These results indicate that DBC is not only robust to perception noise but also scalable with respect to team size, making it well suited for larger heterogeneous teams operating in uncertain environments.

### 5.5 Discussion

The simulation results presented in [Section 5.4](#) demonstrate that DBC achieves a stable balance between convergence speed, robustness, and scalability across varying uncertainty levels and team sizes. These results can be explained by the structural properties of the entropy-based weighting mechanism embedded in the consensus process.

A central observation from the performance evaluation ([Section 5.4.1](#)) is that DBC converges slightly slower than EEV under ideal conditions, yet achieves significantly lower residual error than most classical averaging-based methods. This behavior directly results from the continuous certainty weighting. In noise-free scenarios, entropy values decrease rapidly as beliefs become more concentrated. Consequently, certainty values  $\Omega_i$  increase, increasing the relative influence of concentrated beliefs in the fusion step and accelerating convergence. However, unlike extremum-based methods, DBC does not enforce binary commitment once a threshold is exceeded. Instead, convergence emerges progressively through convex fusion, preventing overshoot and preserving probabilistic consistency. This explains why DBC achieves near-optimal accuracy while maintaining stable dynamics.

The robustness analysis ([Section 5.4.2](#)) reveals a more pronounced structural difference. As the False Discovery Rate (FDR) increases, methods such as LCP and CDCI exhibit rapidly growing consensus error. These algorithms treat observations either equally or through state-based transitions, without explicitly quantifying uncertainty. False positives therefore propagate proportionally through the network and accumulate. In contrast, DBC attenuates unreliable information automatically. In the implemented perception model, injected false detections tend to yield less concentrated and less consistent belief distributions, which increases entropy and lowers certainty values. Reduced certainty reduces their relative influence via the certainty-weighted coefficients  $\omega_j$ , limiting the impact of misleading observations. As a result, error growth remains minimal even at 50% FDR.

EEV demonstrates the opposite dynamic. Its expectation-based extremum rule enforces rapid convergence under clean conditions, but when false positives dominate, incorrect directional bias is amplified rather than damped. This explains the steep error increase observed for EEV at higher FDR levels. DBC avoids this amplification effect because certainty is continuous and adaptive rather than binary.

The scalability results ([Section 5.4.3](#)) further illustrate the structural advantages of local certainty-weighted fusion. As team size increases, classical averaging mechanisms accumulate noise contributions from a larger number of agents, leading to higher final error. DBC, however, scales influence by local certainty and neighborhood structure. Since aggregation depends on  $|N_i|$  rather than the global team size  $N$ , increasing team size does not proportionally increase instability. Larger teams introduce more information,

but only high-certainty contributions significantly shape the consensus, which explains the moderate growth in convergence time and controlled error increase.

An important practical aspect concerns communication uncertainty, particularly packet loss. While the evaluation focuses on asynchronous communication and latency effects, a dedicated sweep over packet loss rates and network partitioning is left for future work. DBC operates asynchronously and does not require synchronized global updates. If messages from certain neighbors are lost, the aggregation step is performed using the subset of available beliefs. The convex normalization in (7)–(9) remains valid for any subset of  $N_i$ , ensuring mathematical consistency under partial communication. High packet loss primarily reduces update frequency rather than altering the update rule itself. In addition, packet loss may lead to temporarily inconsistent beliefs across robots. Such inconsistencies increase entropy, thereby lowering certainty and automatically dampening outdated or conflicting information. This entropy-driven attenuation is expected to contribute to stability in lossy networks. However, prolonged network partitioning or complete isolation of subgroups may delay convergence, as information exchange becomes structurally limited.

### 5.6 Advantages and Limitations

The DBC algorithm offers several notable advantages. First, it integrates a mathematically grounded uncertainty measure directly into the consensus mechanism, enabling adaptive influence modulation without predefined robustness parameters. Second, the method operates in a fully decentralized and asynchronous manner, supporting scalability in heterogeneous robot teams. Third, experimental results demonstrate a favorable balance between convergence speed, robustness to false detections, and scalability with increasing team size. These properties make DBC a promising candidate for decentralized collective perception under uncertainty.

Despite these strengths, several limitations must be critically examined, particularly with respect to real-world deployment. A primary limitation concerns the evaluation environment. The validation of DBC has been conducted exclusively within a controlled simulation. Although the simulation incorporates stochastic sensor noise, communication delays, and false detections, it cannot fully reproduce the chaotic and nonstationary characteristics of physical hardware systems. Real robots experience additional disturbances such as calibration drift, vibration-induced perception errors, temperature-dependent sensor degradation, and correlated noise patterns. Furthermore, wireless communication in field environments may exhibit burst packet loss, asymmetric link quality, or temporary network partitioning, phenomena that are difficult to approximate with simple stochastic delay models. Consequently, while the simulation demonstrates structural robustness, it does not guarantee identical performance under complex physical conditions.

A second limitation relates to correlated and systematic errors. DBC attenuates uncertainty by lowering the influence of high-entropy belief distributions. However, if multiple robots share a systematic bias—such as identical miscalibrated perception models or training data biases—incorrect beliefs may still exhibit low entropy and therefore high certainty. In such cases, DBC may reinforce consistent but wrong interpretations, as the entropy measure captures dispersion rather than correctness. This limitation is particularly relevant in heterogeneous systems integrating similar AI pipelines. A related limitation is probability calibration. If local perception models are overconfident, predictive entropy may underestimate uncertainty, causing  $\Omega$  to overweight unreliable beliefs. Incorporating calibration procedures or uncertainty estimates beyond softmax entropy is therefore an important extension.

Third, the communication model assumes intermittent but generally available connectivity. While DBC tolerates packet loss by operating on partial neighborhood information, prolonged network partitioning may delay or prevent global consensus. If subgroups remain disconnected for extended periods, independent local consensus may form that require additional reconciliation mechanisms once communication is restored.

Finally, computational and bandwidth demands increase with the number of maintained objects and class probabilities. The continuous exchange of full belief distributions and certainty values may strain resource-constrained platforms, particularly in larger teams or multi-class scenarios. Entropy scaling further implies decreasing certainty values as the number of classes grows, potentially slowing convergence in high-dimensional classification tasks.

In summary, while DBC provides structural robustness under modeled uncertainty, comprehensive real-world validation with physical robot platforms remains essential. Future work should therefore include hardware experiments, stress testing under burst communication failures, and analysis under correlated perception biases to fully assess operational robustness beyond simulation.

## 6 Conclusion and Future Work

This paper addressed the challenge of decentralized consensus formation under uncertainty in heterogeneous multi-robot systems. In real-world deployments, autonomous robot teams must generate consistent shared context models despite incomplete sensing, asynchronous communication, and varying levels of reliability across platforms. Robust collective perception is therefore a fundamental prerequisite for cooperative decision-making in domains such as search and rescue, mine detection, infrastructure inspection, and defense-related operations.

To meet this challenge, the DBC algorithm was introduced. The core idea of DBC is the integration of an entropy-based certainty measure directly into the belief aggregation process. By transforming Shannon entropy into a bounded certainty value, each robot adaptively modulates its influence according to the reliability of its local belief. This results in a decentralized, asynchronous, and scalable consensus mechanism that balances convergence speed with robustness against uncertain or misleading information. The experimental evaluation demonstrated that DBC achieves stable and reliable performance across varying uncertainty levels and increasing team sizes, comparing favorably to established consensus algorithms.

From a methodological perspective, future work should extend DBC beyond binary classification settings toward richer multi-class and multi-attribute context models. As the number of classes increases, entropy scaling may reduce discriminative certainty resolution. Investigating refined uncertainty metrics or hybrid confidence measures could enhance robustness in higher-dimensional belief spaces. In addition, adaptive communication strategies—such as event-triggered information exchange or certainty-based transmission thresholds—should be explored to reduce bandwidth requirements in communication-constrained environments. Further methodological extensions include addressing calibration and correlated model bias.

A central priority is real-world validation. In a planned experiment, DBC will be deployed on a heterogeneous team consisting of UAVs and UGVs in an outdoor mine-detection test environment. The setup will emulate varying packet loss rates and sensing conditions to reflect realistic field constraints. The same performance metrics used in simulation—consensus time, consensus error, and consensus rate—will be measured to ensure direct comparability. This experiment will allow systematic assessment of DBC under hardware-induced noise, communication instability, and environmental disturbances, thereby bridging the gap between controlled simulation and operational deployment. Further integration of DBC into broader autonomy architectures also represents a key research avenue. Embedding the consensus mechanism within decentralized task allocation, distributed planning, or learning-based coordination frameworks could enable tighter coupling between collective perception and cooperative action.

In conclusion, DBC provides a principled, entropy-driven approach to decentralized consensus under uncertainty. By explicitly incorporating quantified belief uncertainty into the aggregation process, it advances the robustness and scalability of collective perception. Continued methodological refinement and rigorous

real-world validation will be essential steps toward deploying uncertainty-aware decentralized consensus in practical autonomous robot teams.

**Acknowledgement:** Not applicable.

**Funding Statement:** This research is funded by dtec.bw—Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr (RIVA Project). dtec.bw is funded by the European Union—NextGenerationEU.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization: Marvin Zager and Gianluca Manca; Methodology, software, validation, formal analysis, and investigation: Marvin Zager; Writing—original draft preparation: Marvin Zager and Gianluca Manca; Writing—review and editing: Gianluca Manca, Alexander Fay and Felix Gehlhoff. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are openly available in a Github repository at [https://github.com/hsu-aut/RIVA\\_Decimalized\\_Consensus](https://github.com/hsu-aut/RIVA_Decimalized_Consensus).

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Boateng SN, Singh S, Ugur M, Wang S, Kramer M, Osei I, et al. Heterogenous collaboration: a new approach for search and rescue operations. In: Proceedings of the 2024 IEEE International Symposium on Safety Security Rescue Robotics (SSRR); 2024 Nov 12–14; New York, NY, USA. p. 142–7. doi:10.1109/ssrr62954.2024.10770037.
2. Surmann H, Slomma D, Grobelny S, Grafe R. Deployment of Aerial Robots after a major fire of an industrial hall with hazardous substances, a report. In: Proceedings of the 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR); 2021 Oct 25–27; New York City, NY, USA. p. 40–7. doi:10.1109/ssrr53300.2021.9597677.
3. Vögele T, Sonsalla R, Dettmann A, Cordes F, Maurus M, Dominguez R, et al. Robotics concepts for future planetary exploration missions. In: Yan XT, Visentin G, editors. Space robotics. Cham, Switzerland: Springer Nature; 2024. p. 483–514. doi:10.1007/978-3-031-39214-6\_21.
4. Schweim A, Zager M, Schweim M, Fay A, Horn J. Unmanned vehicles on the rise: a review on projects of cooperating robot teams. *Autom.* 2024;72(1):3–14. doi:10.1515/auto-2022-0153.
5. Da Vieira Silva LM, Köcher A, Fay A. A capability and skill model for heterogeneous autonomous robots. *Autom.* 2023;71(2):140–50. doi:10.1515/auto-2022-0122.
6. Amirkhani A, Barshooi AH. Consensus in multi-agent systems: a review. *Artif Intell Rev.* 2022;55(5):3897–935. doi:10.1007/s10462-021-10097-x.
7. Yang F, Din A, Liu H, Babar M, Ahmad S. Decentralized consensus in robotic swarm for collective collision and avoidance. *IEEE Access.* 2024;12(23):72143–54. doi:10.1109/access.2024.3402564.
8. Moussa M, Beltrame G. On the robustness of consensus-based behaviors for robot swarms. *Swarm Intell.* 2020;14(3):205–31. doi:10.1007/s11721-020-00183-1.
9. Prorok A, Malencia M, Carlone L, Sukhatme GS, Sadler BM, Kumar V. Beyond robustness: a taxonomy of approaches towards resilient multi-robot systems. arXiv:2109.12343. 2021. doi:10.48550/arXiv.2109.12343.
10. Gao P, Zhu Q, Zhang H. Uncertainty-aware correspondence identification for collaborative perception. *Auton Rob.* 2023;47(5):635–48. doi:10.1007/s10514-023-10086-9.
11. Ullah I, Adhikari D, Khan H, Anwar MS, Ahmad S, Bai X. Mobile robot localization: current challenges and future prospective. *Comput Sci Rev.* 2024;53(8):100651. doi:10.1016/j.cosrev.2024.100651.
12. Balestrieri E, Daponte P, De Vito L, Lamonaca F. Sensors and measurements for unmanned systems: an overview. *Sensors.* 2021;21(4):1518. doi:10.3390/s21041518.

13. Li Z, Chen J. Robust consensus for multi-agent systems communicating over stochastic uncertain networks. *SIAM J Control Optim.* 2019;57(5):3553–70. doi:10.1137/18m1181614.
14. Choudhury S, Gupta JK, Kochenderfer MJ, Sadigh D, Bohg J. Dynamic multi-robot task allocation under uncertainty and temporal constraints. *Auton Rob.* 2022;46(1):231–47. doi:10.1007/s10514-021-10022-9.
15. Luo W, Sun W, Kapoor A. Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates. *Adv Neural Inf Process Syst.* 2020;33:372–83.
16. Mujkic E, Ravn O, Christiansen MP. Framework for environment perception: ensemble method for vision-based scene understanding algorithms in agriculture. *Front Robot AI.* 2023;9:982581. doi:10.3389/frobt.2022.982581.
17. Mengers V, Raoufi M, Brock O, Hamann H, Romanczuk P. Leveraging uncertainty in collective opinion dynamics with heterogeneity. *Sci Rep.* 2024;14(1):27314. doi:10.1038/s41598-024-78856-8.
18. Shan Q, Mostaghim S. Discrete collective estimation in swarm robotics with distributed Bayesian belief sharing. *Swarm Intell.* 2021;15(4):377–402. doi:10.1007/s11721-021-00201-w.
19. Vivoli E, Bertini M, Capineri L. Deep learning-based real-time detection of surface landmines using optical imaging. *Remote Sens.* 2024;16(4):677. doi:10.3390/rs16040677.
20. Kochenderfer MJ. *Decision making under uncertainty: theory and application.* Cambridge, MA, USA: The MIT Press; 2015.
21. Zhou L, Tokekar P. Multi-robot coordination and planning in uncertain and adversarial environments. *Curr Robot Rep.* 2021;2(2):147–57. doi:10.1007/s43154-021-00046-5.
22. Reina A, Valentini G, Fernández-Oto C, Dorigo M, Trianni V. A design pattern for decentralised decision making. *PLoS One.* 2015;10(10):e0140950. doi:10.1371/journal.pone.0140950.
23. Shan Q, Mostaghim S. Many-option collective decision making: discrete collective estimation in large decision spaces. *Swarm Intell.* 2024;18(2):215–41. doi:10.1007/s11721-024-00239-6.
24. Acar E, Bayrak G, Jung Y, Lee I, Ramu P, Ravichandran SS. Modeling, analysis, and optimization under uncertainties: a review. *Struct Multidiscip Optim.* 2021;64(5):2909–45. doi:10.1007/s00158-021-03026-7.
25. Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Automat Contr.* 2004;49(9):1520–33. doi:10.1109/tac.2004.834113.
26. Zhang G, Pan F, Mao Y, Tijanic S, Dang'ana M, Motepalli S, et al. Reaching consensus in the Byzantine empire: a comprehensive review of BFT consensus algorithms. *ACM Comput Surv.* 2024;56(5):1–41. doi:10.1145/3636553.
27. Adoni W, Lorenz S, Fareedh J, Gloaguen R, Bussmann M. Investigation of autonomous multi-UAV systems for target detection in distributed environment: current developments and open challenges. *Drones.* 2023;7(4):263. doi:10.3390/drones7040263.
28. Ji JC, Lam CT, Ng B. A survey on consensus algorithms for distributed wireless networks. *The international conference optoelectronic information.* 2025 [cited 2026 Jan 1]. Available from: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/13513/3045449/A-survey-on-consensus-algorithms-for-distributed-wireless-networks/10.1117/12.3045449.full>.
29. Arvin F, Turgut AE, Yue S. Fuzzy-based aggregation with a mobile robot swarm. In: *Swarm intelligence.* Berlin/Heidelberg, Germany: Springer; 2012. p. 346–7. doi:10.1007/978-3-642-32650-9\_39.
30. Pan M, Yang Y, Qin X, Liu L. Self-organized aggregation with physical interaction. In: *Proceedings of the 2023 IEEE International Conference on Real-time Computing and Robotics (RCAR); 2023 Jul 17–20; Datong, China.* p. 182–7. doi:10.1109/rcar58764.2023.10249716.
31. Karagüzel TA, Turgut AE, Eiben AE, Ferrante E. Collective gradient perception with a flying robot swarm. *Swarm Intell.* 2023;17(1):117–46. doi:10.1007/s11721-022-00220-1.
32. Vega R, Nowzari C. Classifying emergence in robot swarms: an observer-dependent approach. *arXiv:2507.07315.* 2015. doi:10.48550/arXiv.2507.07315.
33. Salman M, Garzón Ramos D, Birattari M. Automatic design of stigmergy-based behaviours for robot swarms. *Commun Eng.* 2024;3(1):30. doi:10.1038/s44172-024-00175-7.
34. Huo X, Zhang H, Wang Z, Huang C, Yan H. A self-learning approach to heterogeneous multi-robot coalition formation under uncertainty. *IEEE Trans Automat Sci Eng.* 2025;22(6):3445–57. doi:10.1109/tase.2024.3395283.

35. Street C, Lacerda B, Mühlhig M, Hawes N. Right place, right time: proactive multi-robot task allocation under spatiotemporal uncertainty. *J Artif Intell Res.* 2024;79:137–71. doi:10.1613/jair.1.15057.
36. Bi W, Shen J, Zhou J, Zhang A. Heterogeneous multi-UAV mission reallocation based on improved consensus-based bundle algorithm. *Drones.* 2024;8(8):345. doi:10.3390/drones8080345.
37. Yoon S, Do H, Kim J. Robust task allocation for multiple cooperative robotic vehicles considering node position uncertainty. *J Intell Rob Syst.* 2022;106(1):23. doi:10.1007/s10846-022-01732-y.
38. Dhanaraj N, Kang JH, Mukherjee A, Nemlekar H, Nikolaidis S, Gupta SK. Multi-robot task allocation under uncertainty via hindsight optimization. In: *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*; 2024 May 13–17; Yokohama, Japan. p. 16574–80. doi:10.1109/icra57147.2024.10611370.
39. LeBlanc HJ, Zhang H, Koutsoukos X, Sundaram S. Resilient asymptotic consensus in robust networks. *IEEE J Select Areas Commun.* 2013;31(4):766–81. doi:10.1109/jsac.2013.130413.
40. Shan Q, Mostaghim S. Noise-resistant and scalable collective preference learning via ranked voting in swarm robotics. *Swarm Intell.* 2023;17(1):5–26. doi:10.1007/s11721-022-00214-z.
41. Saari DG. The optimal ranking method is the borda count. Discussion paper 1985; No. 638. [cited 2026 Jan 1]. Available from: <https://hdl.handle.net/10419/220997>.
42. Abdelli A, Yachir A, Amamra A, Khaldi B. Maximum likelihood estimate sharing for collective perception in static environments for swarm robotics. *Robotica.* 2023;41(9):2754–73. doi:10.1017/s0263574723000668.
43. Shannon CE. A mathematical theory of communication. *Bell Syst Tech J.* 1948;27(4):623–56. doi:10.1002/j.1538-7305.1948.tb00917.x.
44. Vuong QH, La VP, Nguyen MH. Informational entropy-based value formation: a new paradigm for a deeper understanding of value. *SSRN J.* 2025;13(3):166. doi:10.2139/ssrn.5126652.
45. Yalcin E, Ismailoglu F, Bilge A. An entropy empowered hybridized aggregation technique for group recommender systems. *Expert Syst Appl.* 2021;166:114111. doi:10.1016/j.eswa.2020.114111.
46. Dombi J, Fáró J, Jónás T. A fuzzy entropy-based group consensus measure for financial investments. *Mathematics.* 2024;12(1):4. doi:10.3390/math12010004.
47. Geide-Stevenson D, La Parra-Pérez Á. Consensus among economists 2020—a sharpening of the picture. *J Econ Educ.* 2024;55(4):461–78. doi:10.1080/00220485.2024.2386328.
48. Cheng J, Xu G, Yuan G, Yang L, Huang Z, Huang C, et al. A side chain consensus-based decentralized autonomous vehicle group formation and maintenance method in a highway scene. *IEEE Trans Ind Inf.* 2022;18(12):9250–8. doi:10.1109/tii.2022.3178427.
49. Liu X, Hang P, Wang Y, Sun J. A cooperative decision-making method for CAVs from the perspective of opinion dynamics. *Transp Res Part C Emerg Technol.* 2026;182(1):105412. doi:10.1016/j.trc.2025.105412.
50. Sharma G, Ahmed R, Patel S, Mishra P, Azam M, Singh S. GPS integrated landmine detection rover with camera. In: *Proceedings of the 2024 International Conference on Signal Processing and Advance Research in Computing (SPARC)*; 2024 Sep 12–13; Lucknow, India. p. 1–6. doi:10.1109/sparc61891.2024.10829152.