Quantitative 4-Valued Information Aggregation in Multi-Agent Systems

Alessandro Aldini[1], Pierluigi Graziani[1], Claudio Antares Mezzina[2], and Gandolfo Vergottini[1]

[1] University of Urbino Carlo Bo, Italy
[2] University of Bari Aldo Moro, Italy

Abstract. Multi-agent systems operating under incomplete or conflicting information require aggregation mechanisms that preserve distinctions lost in binary representations. We introduce a quantitative operational framework based on four weighted informational labels: Correct (C), Incorrect (I), Undefined (U), and Conflict (B). The framework supports incremental aggregation and guarantees that, for any fixed finite collection of contributions, the resulting qualitative label is independent of processing order. While the complete weighted state is also order-independent for C, I, and U outcomes, a final conflict state may retain an order-dependent provisional weight. We therefore introduce a matrix representation that assigns a canonical conflict weight and yields a unique order-independent normalized state. The framework is further specialized to a hierarchical leader–follower architecture with post-aggregation corrective feedback. Under explicit delivery assumptions, we prove finite termination and qualitative alignment with the normalized collective assessment. The approach is illustrated through a multi-agent search and rescue scenario.

1 Introduction

Multi-agent systems operating in dynamic environments frequently make decisions under incomplete, unavailable, or contradictory local information. While standard consensus models often rely on binary [11, 10] or probabilistic frameworks [19], purely binary representations fail to distinguish between a complete absence of information and unresolved opposition [6, 22].

To address this issue, we propose a formal, logic-inspired framework for quantitative information aggregation grounded in 4-valued semantics. We model four operational epistemic states: Correct (C, supporting a proposition ϕ), Incorrect (I, opposing ϕ), Undefined (U, absence of usable assessment), and Conflict (B, unresolved opposition between determinate contributions). Local reports carry unit weights, which accumulate or cancel out during aggregation to yield a weighted collective state. We formally prove that the qualitative result of this incremental aggregation is invariant to the temporal order of report processing, and introduce a canonical matrix representation to ensure order-independent state normalization.

We specialize this general framework to a hierarchical leader–follower setting. In this architecture, followers provide unit-weight assessments to a leader, which computes the normalized collective state. If this state is determinate (C or I), a post-aggregation feedback mechanism—formalized via *Propagation of Retractions* rules—sends corrective feedback to non-aligned or undefined followers. Under explicit message delivery assumptions, we prove that this corrective phase terminates in finite steps and achieves qualitative alignment across the hierarchy.

The primary contribution of this work is the formal integration of 4-valued epistemic states, incremental order-independent aggregation, and finite corrective feedback within a multi-agent framework. We demonstrate these formal mechanisms using a multi-agent search and rescue example.

The paper is organized as follows. Section 2 reviews related work in distributed aggregation and formal consensus. Section 3 outlines the temporal structure of communication rounds. Section 4 details the operational rules, order-invariance proofs, matrix normalization, and hierarchical feedback specialization. Section 5 presents the search and rescue example, and Section 6 concludes with future directions.

2 Related Work and Positioning

Our framework intersects distributed algorithms, population protocols, and multi-valued logics, but occupies a distinct operational position focused on quantitative, order-independent information aggregation.

Classical distributed computing studies coordination and agreement under different synchrony and fault assumptions [16]. Population protocols compute global predicates through pairwise interactions [3]. On primitive reports, our qualitative outcome is likewise determined by counting conditions; our focus is their integration with weighted informational states and post-aggregation corrective feedback. On the determinate fragment, aggregation reduces to a signed balance. The balance alone does not distinguish absence of determinate information (U) from balanced opposition (B). Our framework represents this distinction explicitly and provides a canonical normalization of the completed round.

Information fusion under inconsistency is widely studied in belief revision and merging [6, 22]. Our four informational labels (C, I, U, B) are directly motivated by Belnap–Dunn 4-valued semantics. However, we treat these labels as operational information states rather than logical truth values, and our aggregation operator $\star$ as an operational update rule rather than a truth-functional connective. Rather than performing belief revision over complex formula sets, our model provides a lightweight, rule-based operational semantics for incremental multi-agent aggregation, bridging symbolic multi-valued logic with quantitative, state-based interaction models.

3 Operational Model and Temporal Dynamics

An aggregation round operates over a fixed set of local contributions to determine a stabilized collective state. The system evolves through a finite sequence of discrete temporal phases:

- Time t_0 (Initial State): The system rests in a steady baseline state.
- Time t_1 (Measurement): Agents perform local sensing during a prescribed interval.
- Time t_2 (Local Assessment): Each agent assigns a unit-weight label to proposition ϕ: C_1 (supporting) or I_1 (opposing). If a sensing failure or *measurement timeout* occurs, the agent assigns U_1 (undefined).
- Time t_3 (Exchange): Agents transmit/receive the assigned labels. If a message fails to arrive within the communication window (*communication timeout*), the receiver records a default report U_1 for the missing sender.
- Time t_4 (Sequential & Matrix Aggregation): Each receiving agent sequentially incorporates the contributions available for the round via the operational rules (see Section 4.1). Upon incorporating all round contributions, the agent computes the corresponding matrix representation to adopt a normalized, order-independent final state (see Section 4.2).
- Time t_5 (Post-Aggregation Feedback): If a normalized state is determinate (C_y or I_y), it serves as the reference state for a corrective feedback phase. Non-determinate states (U, B) trigger no action. No new sensing occurs, leaving the normalized round outcome unchanged.

We define the sequence t_0–t_4 as the *communication round*, and t_5 as the *post-aggregation feedback phase*.

4 Operational Rules and Order-Invariant Aggregation

Following the temporal structure of Section 3, we define the operational rules used at time t_4 to aggregate information. The framework represents assessments of a proposition ϕ using four operational labels[3]: Correct (C, supporting ϕ), Incorrect (I, opposing ϕ), Undefined (U, absence of usable assessment), and Conflict (B, unresolved opposition).

A weighted informational state for agent N_i is denoted $\delta_x(\phi_{N_i})$ (abbreviated $\delta_x(N_i)$ when ϕ is clear from the context, or $\delta(N_i)$ when $x = 1$), where $\delta \in \{U, I, C, B\}$ and $x \in \mathbb{N}_{>0}$.

Local assessments assigned at t_2 have unit weight ($x = 1$), yielding initial states C_1, I_1, or U_1. The conflict label B arises solely through aggregation.

When agent N_i (in state δ_x) transmits its report to agent N_j (in state γ_y) over directed channel $\overset{i\to j}{R}(N_i, N_j)$, the receiving agent updates its state to $\lambda_z(N_j)$ via inference rules of the form:

[3] Inspired by Belnap–Dunn semantics [5, 6], U and B represent incomplete and inconsistent information, respectively. Here, however, they serve strictly as operational aggregation states.

$$\frac{\delta_x(N_i) \qquad \gamma_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{\lambda_z(N_j)}$$

4.1 The Operational Rules

The following operational rules govern the information aggregation at time t_4. They are divided into three groups and, for all of them, we assume that the informational states refer to the same base proposition.

Non-determinate interactions. Rules $[1_A]$–$[2_B]$ govern interactions between non-determinate states (U and B). Rules $[1_A]$ and $[1_B]$ combine weights when identical non-determinate states interact, whereas rules $[2_A]$ and $[2_B]$ ensure that undefined information (U) never resolves an existing conflict state (B), preserving B and accumulating total weights.

$$\frac{U_x(N_i) \qquad U_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{U_{x+y}(N_j)}\ [1_A] \qquad \frac{B_x(N_i) \qquad B_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{B_{x+y}(N_j)}\ [1_B]$$

$$\frac{B_x(N_i) \qquad U_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{B_{x+y}(N_j)}\ [2_A] \qquad \frac{U_x(N_i) \qquad B_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{B_{x+y}(N_j)}\ [2_B]$$

Determinate interactions. Rules $[3_A]$-$[3_B]$ and rules $[8_A]$-$[8_B]$ govern interactions between determinate states (C and I). Aligning assessments accumulate weights ($[3_A]$, $[3_B]$), whereas opposing assessments cancel out ($[8_A]$, $[8_B]$). For opposing interactions, the dominant state survives with weight $z = |x - y|$; if $x = y$, opposite contributions perfectly balance to yield conflict state B_x.

$$\frac{C_x(N_i) \qquad C_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{C_{x+y}(N_j)}\ [3_A] \qquad \frac{I_x(N_i) \qquad I_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{I_{x+y}(N_j)}\ [3_B]$$

$$\frac{C_x(N_i) \qquad I_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{\delta_z(N_j)}\ [8_A] \qquad \delta = \begin{cases} I, & \text{if } x < y \text{ and } z = y - x, \\ C, & \text{if } y < x \text{ and } z = x - y, \\ B, & \text{if } x = y \text{ and } z = x = y. \end{cases}$$

$$\frac{I_x(N_i) \qquad C_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{\delta_z(N_j)}\ [8_B] \qquad \delta = \begin{cases} C, & \text{if } x < y \text{ and } z = y - x, \\ I, & \text{if } y < x \text{ and } z = x - y, \\ B, & \text{if } x = y \text{ and } z = x = y. \end{cases}$$

Mixed interactions. Rules $[4_A]$–$[7_B]$ govern interactions between determinate (C, I) and non-determinate (U, B) states. Determinate states take operational precedence: incoming C or I contributions overwrite an existing U or B state ($[4_A]$, $[4_B]$, $[7_A]$, $[7_B]$), whereas non-determinate reports leave an existing C or I state and its weight unchanged ($[5_A]$, $[5_B]$, $[6_A]$, $[6_B]$). Non-determinate weights measure accumulated non-determinate history rather than commensurable evidence scores; thus, U and B do not weaken or subtract from a determinate balance.

$$\frac{C_x(N_i) \qquad U_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{C_x(N_j)}\ [4_A] \qquad \frac{I_x(N_i) \qquad U_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{I_x(N_j)}\ [4_B]$$

$$\frac{B_x(N_i) \qquad C_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{C_y(N_j)}\ [5_A] \qquad \frac{B_x(N_i) \qquad I_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{I_y(N_j)}\ [5_B]$$

$$\frac{U_x(N_i) \qquad C_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{C_y(N_j)}\ [6_A] \qquad \frac{U_x(N_i) \qquad I_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{I_y(N_j)}\ [6_B]$$

$$\frac{C_x(N_i) \qquad B_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{C_x(N_j)}\ [7_A] \qquad \frac{I_x(N_i) \qquad B_y(N_j) \qquad \overset{i\to j}{R}(N_i, N_j)}{I_x(N_j)}\ [7_B]$$

Aggregated weights represent accumulated or residual support rather than probabilities, confidence levels, or agent reliability scores. For instance, C_3 can result either from three positive evaluations with a weight of one each, or from five positive and two negative reports: its weight is a net balance, not a confidence level. While determinate (C, I) and undefined (U) weights represent straightforward additive balances, the weight of a provisional conflict state (B) generated sequentially may depend on the message processing order. To eliminate this residual order-dependence and assign a canonical conflict weight, we introduce an order-independent matrix representation in Section 4.2.

4.2 Order Invariance and Matrix Normalization

In multi-agent systems, information arrives at every agent asynchronously and is processed sequentially. Fixing a receiving agent N_j and a given communication round, let us denote with

$$\Gamma = (\alpha_1, \ldots, \alpha_n) \in \mathcal{W}^n$$

the family of $n \geq 1$ weighted contributions available to N_j in that round. Each α_i is the informational state carried by one contribution to be processed by the receiver. In the following, for simplicity, we suppress agent identities, since we focus on the combination of informational states and the resulting final qualitative state, which we will show to be independent of the specific order with which the α_i are processed. Formally,

$$\mathcal{W} = \{\delta_x : \delta \in \{U, I, C, B\},\ x \in \mathbb{N}_{>0}\}$$

is the set of weighted informational states, with $\mathrm{lab}(\delta_x) = \delta$ and $\mathrm{wt}(\delta_x) = x$. Primitive assessments generated by local measurements or timeouts belong to $\mathcal{P} = \{C_1, I_1, U_1\} \subset \mathcal{W}$.

Moreover, we write $\alpha \star \beta$ for the weighted state obtained by applying rules $[1_A]$–$[8_B]$ to an incoming contribution α and the current state β of the fixed receiver. For any permutation π of $\{1, \ldots, n\}$, the sequential evaluation of Γ is defined inductively by:

$$E_1^\pi = \alpha_{\pi(1)}, \qquad E_{m+1}^\pi = \alpha_{\pi(m+1)} \star E_m^\pi \quad (1 \leq m < n), \qquad E^\pi(\Gamma) = E_n^\pi$$

where $E^\pi(\Gamma)$ is the final state of the evaluation. To analyze structural invariants, we define the net balance $\sigma : \mathcal{W} \to \mathbb{Z}$ and non-undefined indicator $\kappa : \mathcal{W} \to \{0, 1\}$:

$$\sigma(C_x) = x, \quad \sigma(I_x) = -x, \quad \sigma(U_x) = \sigma(B_x) = 0,$$
$$\kappa(U_x) = 0, \quad \kappa(C_x) = \kappa(I_x) = \kappa(B_x) = 1.$$

The value $\sigma(\alpha)$ represents the net determinate balance of α: a state C_x contributes x, a state I_x contributes $-x$, while U_x and B_x contribute no net determinate balance. The value $\kappa(\alpha)$ distinguishes an undefined state from any non-undefined state: it is 0 exactly on U-states and 1 on C-, I-, and B-states.

Lemma 1. *For all $\alpha, \beta \in \mathcal{W}$:*

- $\sigma(\alpha \star \beta) = \sigma(\alpha) + \sigma(\beta)$;
- $\kappa(\alpha \star \beta) = \max\{\kappa(\alpha), \kappa(\beta)\}$.

Consequently, for every non-empty finite family Γ and processing order π:

$$\sigma\big(E^\pi(\Gamma)\big) = \sum_{r=1}^{n} \sigma(\alpha_r) \qquad \textit{and} \qquad \kappa\big(E^\pi(\Gamma)\big) = \max_{1 \le r \le n} \kappa(\alpha_r).$$

Proof. Regarding $\sigma(\alpha \star \beta)$, the proof follows by inspection of rules $[1_A]$–$[8_B]$. Determinate rules add (see rules $[3_A]$ and $[3_B]$) or subtract (see rules $[8_A]$ and $[8_B]$) balances linearly, while non-determinate and mixed interactions preserve balances.

Regarding $\kappa(\alpha \star \beta)$, rule $[1_A]$ ($U \star U \to U$) is the unique rule yielding 0; every interaction involving at least one C-, I-, or B-state produces a state different from U. Hence, the non-undefined indicator of the result is the maximum of the indicators of the two premises.

The last two equalities follow by induction on the number of elements of Γ. The base case $n = 1$ is trivial. For the induction step, assume the claim for the first m contributions in the order π. Applying the two binary equalities to $\alpha_{\pi(m+1)}$ and E^π_m yields the required sum and maximum for $m+1$ contributions. □

In the following, we write $\Sigma(\Gamma) = \sum_{r=1}^{n} \sigma(\alpha_r)$ and $K(\Gamma) = \max_{1 \le r \le n} \kappa(\alpha_r)$.

Lemma 2. *For every non-empty finite family Γ and processing order π:*

$$\mathrm{lab}\big(E^\pi(\Gamma)\big) = \begin{cases} C, & \textit{if } \Sigma(\Gamma) > 0, \\ I, & \textit{if } \Sigma(\Gamma) < 0, \\ U, & \textit{if } \Sigma(\Gamma) = 0 \textit{ and } K(\Gamma) = 0, \\ B, & \textit{if } \Sigma(\Gamma) = 0 \textit{ and } K(\Gamma) = 1. \end{cases}$$

Proof. Positive and negative balances correspond uniquely to labels C and I. Zero-balance states are partitioned by $K(\Gamma)$: indicator 0 yields U, while indicator 1 yields B. The result follows directly from Lemma 1. □

The following theorem states that the qualitative label obtained by applying the operational rules is independent of the order in which the contributions are processed.

Theorem 1 (Qualitative Order Invariance). *For every non-empty finite family Γ and for any two processing orders π and τ,* $\mathrm{lab}\big(E^{\pi}(\Gamma)\big) = \mathrm{lab}\big(E^{\tau}(\Gamma)\big)$.

Proof. $\Sigma(\Gamma)$ and $K(\Gamma)$ are symmetric functions of Γ. By Lemma 2, they uniquely determine the qualitative label independently of evaluation order. □

The preceding invariants imply weight uniqueness for C and I: $E^{\pi}(\Gamma)$ yields $C_{\Sigma(\Gamma)}$ if $\Sigma(\Gamma) > 0$ and $I_{-\Sigma(\Gamma)}$ if $\Sigma(\Gamma) < 0$. For U, $K(\Gamma) = 0$ forces every contribution to be a U-state, and repeated applications of rule $[1_A]$ yield $U_{\sum \mathrm{wt}(\alpha_r)}$.

However, the qualitative invariance of the order does not extend to the provisional weights of the conflicts. This difference stems from the treatment of indefinite information: $U_1 \star I_1 = I_1$, while $U_1 \star B_1 = B_2$. Consequently, $\star$ is not associative on weighted states: $(U_1 \star I_1) \star C_1 = B_1$, whereas $U_1 \star (I_1 \star C_1) = B_2$. Both evaluations, however, have the same qualitative label B, as guaranteed by Theorem 1.

To assign a canonical conflict weight, we define an order-independent matrix associated with $\Gamma = (\alpha_1, \ldots, \alpha_n)$ as follows:

$$M(\Gamma) = \begin{pmatrix} x_U & y_I \\ z_C & v_B \end{pmatrix}$$

where x, y, z, v sum the weights of all U, I, C, and B contributions in Γ, respectively (with entry 0 indicating absence of contributions):

$$x = \sum_{\substack{1 \le r \le n \\ \mathrm{lab}(\alpha_r) = U}} \mathrm{wt}(\alpha_r) \quad y = \sum_{\substack{1 \le r \le n \\ \mathrm{lab}(\alpha_r) = I}} \mathrm{wt}(\alpha_r) \quad z = \sum_{\substack{1 \le r \le n \\ \mathrm{lab}(\alpha_r) = C}} \mathrm{wt}(\alpha_r) \quad v = \sum_{\substack{1 \le r \le n \\ \mathrm{lab}(\alpha_r) = B}} \mathrm{wt}(\alpha_r)$$

Definition 1. *The canonical evaluation $\rho(M(\Gamma))$ of the matrix $M(\Gamma)$ associated with a non-empty finite family Γ of informational contributions is defined by:*

$$\rho(M(\Gamma)) = \begin{cases} U_x, & \text{if } x > 0 \text{ and } y = z = v = 0, \\ C_{z-y}, & \text{if } z > y, \\ I_{y-z}, & \text{if } y > z, \\ B_{x+v+y}, & \text{if } y = z \text{ and } y + v > 0. \end{cases}$$

The canonical evaluation $\rho(M(\Gamma))$ is computed from the matrix of the complete family of contributions and, as we will see, need not coincide with the weighted final state $E^{\pi}(\Gamma)$ of any sequential evaluation of Γ.

The canonical conflict weight $x + v + y$ has an accounting interpretation over the complete family of contributions. When $y = z > 0$, the total determinate weights correspond to C_y and I_y, and rules $[8_A]$–$[8_B]$ yield $C_y \star I_y = B_y$. Thus, y records balanced determinate opposition, while x and v account for undefined

and already conflicting contributions. Hence, $x+v+y$ is a canonical representation of the completed round, not necessarily the result of a sequential evaluation. For example, every sequential evaluation of (C_1, C_1, I_1, I_1) yields B_1: immediately before the last contribution is processed, the accumulated state has the opposite determinate label and weight 1. By contrast, the matrix has $x = v = 0$ and $y = z = 2$, so Definition 1 gives B_2.

The following theorem establishes qualitative correspondence, with equality of weighted states for outcomes labelled C, I, or U.

Theorem 2 (Rules–Matrix Correspondence). *For every non-empty finite family Γ and processing order π, $\mathrm{lab}\big(E^\pi(\Gamma)\big) = \mathrm{lab}\big(\rho(M(\Gamma))\big)$. The two evaluations yield the same weighted state for C, I, U, while for label B, $\rho(M(\Gamma))$ assigns the canonical conflict weight $x + v + y$.*

Proof. By definition, $\Sigma(\Gamma) = z - y$ and $K(\Gamma) = 0 \iff y = z = v = 0$. Hence Lemma 2 and Definition 1 determine the same qualitative label. If $z > y$, both evaluations yield C_{z-y}; if $y > z$, both yield I_{y-z}. If the common label is U, all contributions are U-states and both evaluations yield U_x. In the remaining case, the common label is B, and the matrix assigns the canonical weight $x+v+y$. □

The normalized final state is defined as $\widehat{E}^{\,\pi}(\Gamma) = E^\pi(\Gamma)$ if $\mathrm{lab}\big(E^\pi(\Gamma)\big) \neq B$, and $\rho(M(\Gamma))$ otherwise.

Theorem 3 (Order Invariance and Canonical Normalization). *For every non-empty finite family Γ and any two processing orders π and τ:*

$$\widehat{E}^{\,\pi}(\Gamma) = \widehat{E}^{\,\tau}(\Gamma) = \rho(M(\Gamma)).$$

Proof. Fix a processing order π. If $\mathrm{lab}(E^\pi(\Gamma)) \neq B$, then by Theorem 2, $E^\pi(\Gamma) = \rho(M(\Gamma))$, and hence $\widehat{E}^{\,\pi}(\Gamma) = \rho(M(\Gamma))$. If $\mathrm{lab}(E^\pi(\Gamma)) = B$, the same equality follows directly from the definition of $\widehat{E}^{\,\pi}(\Gamma)$. Thus $\widehat{E}^{\,\pi}(\Gamma) = \rho(M(\Gamma))$ for every processing order π, and the result follows. □

Figure 1 explicitly illustrates the distinction between non-associativity, qualitative order invariance, and matrix normalization for the same family of contributions.

Thus, $E^\pi(\Gamma)$ represents the sequential evaluation of the round, whereas $M(\Gamma)$ retains the complete aggregate information used to compute the order-independent normalized state $\rho(M(\Gamma))$. Notice that the normalized state alone does not retain all the information needed for an exact subsequent canonical update. For example, (C_1, C_1, I_1) and (C_1, U_1, U_1) both normalize to C_1, but appending the same contribution I_1 yields B_2 and B_3, respectively. The matrix representation retains the information lost by the normalized state.

4.3 System Architecture: Hierarchical Multi-Agent Setting

We now specialize the fixed receiving agent N_j of Section 4.2 to a leader N_α, with follower agents $\{N_1, \ldots, N_n\}$ providing unit-weight assessments. For each

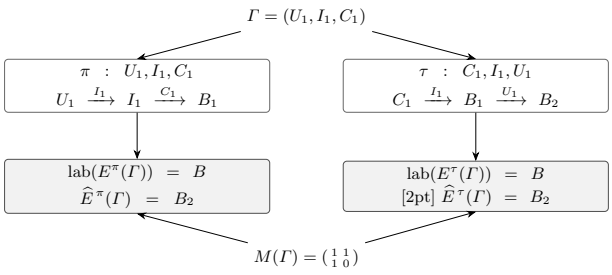

Fig. 1. Different processing orders yield the same conflict label but different provisional weights; normalization uses the full contribution matrix.

follower, the leader records r_i as the received assessment, or U_1 upon a communication timeout.

During the exchange phase, information flows from the follower agents to the leader agent. The leader subsequently processes the recorded contributions during the aggregation phase. Accordingly, in applications of the aggregation rules, the receiving agent N_j is interpreted as the leader. During the retractions phase at time t_5, the direction of communication is reversed, since corrective messages are sent by the leader to selected followers.

The leader maintains the follower record $\mathcal{R}_\alpha = \{(N_i, r_i) : 1 \le i \le n\}$ alongside the family $\Gamma_\alpha = (r_1, \ldots, r_n)$. The record $\mathcal{R}_\alpha$ and the matrix $M(\Gamma_\alpha)$ serve different purposes. The former identifies which contribution is associated with each follower and is therefore used during the retractions phase. The latter discards agent identities and retains only the aggregate weights of the four informational labels; it is used to compute and normalize the collective informational state.

The leader's empty register prior to aggregation is denoted $\emptyset(N_\alpha)$. Initialization is governed by rule $[0_A]$:

$$\frac{r_i = \delta_1 \qquad \emptyset(N_\alpha)}{\delta_1(N_\alpha)}\,[0_A]$$

where $\delta \in \{U, I, C\}$. Subsequent contributions are incorporated using the operational rules, abbreviated by $\star$. Consequently, sequential evaluation on Γ_α matches $E^{\pi}(\Gamma_\alpha)$, and Theorems 1–3 apply directly.

4.4 Propagation of Retractions

The operational rules allow the leader to aggregate the contributions associated with the followers during a communication round. They do not, however, provide feedback to followers whose recorded contributions are undefined or differ from the normalized final assessment of the leader. For this purpose, we introduce the *Propagation of Retractions* rules. The term retraction is used operationally for this post-aggregation corrective step; the rule does not constitute a general

belief-revision operator. More precisely, the message acts as a corrective-priority update relative to the normalized collective state.

We use the notation $[\delta_x(N_i)]^{t_j}$ for a weighted state associated with a time step, where $\delta \in \{U, I, C, B\}$. At time t_3, each follower N_i has a local unit-weight state $s_i \in \{C_1, I_1, U_1\}$, assigned during the local assessment phase at time t_2. The leader also has the recorded contribution $r_i \in \{C_1, I_1, U_1\}$ defined in the preceding subsection. If the follower's report is successfully received, then $r_i = s_i$. If a communication timeout occurs, the leader records $r_i = U_1$, while the actual local state s_i of the non-responding follower may be any of C_1, I_1, or U_1.

The retraction phase is activated only when the normalized state of the leader is a determinate state, namely C_y or I_y. If the leader ends the aggregation phase in state U or B, no retraction rule is applied. A recorded follower contribution is called *discrepant* when it carries the determinate label opposite to that of the leader. This does not by itself establish that the follower is faulty; it records only disagreement with the aggregated assessment. Nor does the normalized assessment establish the external truth of the proposition. The objective of the feedback phase is qualitative alignment with the collective result of the completed round, not truth convergence. In particular, if the majority of determinate measurements is systematically wrong, the mechanism may align a locally correct follower with an externally incorrect collective assessment; such fault and reliability questions lie outside the scope of our results.

The leader uses the per-agent record $\mathcal{R}_\alpha$ to identify the followers whose recorded contributions are undefined or discrepant. The matrix $M(\Gamma_\alpha)$ is not used for this purpose, since it does not retain agent identities. The Propagation of Retractions rules then generate corrective messages for the selected followers:

$$\frac{[r_i = \delta_1]^{t_3} \qquad [C_y(N_\alpha)]^{t_4} \qquad \overset{\alpha \to i}{R}(N_\alpha, N_i)}{\mathrm{msg}^{t_5}_{\alpha \to i}(C_{y+1})}\,[R_1]$$

where $\delta \in \{U, I\}$ and $y \geq 1$, and

$$\frac{[r_i = \delta_1]^{t_3} \qquad [I_y(N_\alpha)]^{t_4} \qquad \overset{\alpha \to i}{R}(N_\alpha, N_i)}{\mathrm{msg}^{t_5}_{\alpha \to i}(I_{y+1})}\,[R_2]$$

where $\delta \in \{U, C\}$ and $y \geq 1$.

The application of rule $[R_1]$ or $[R_2]$ does not modify the normalized state of the leader and does not add a new contribution to the matrix of the completed round. It only generates a corrective message whose content, if successfully delivered, is processed against the current local state of the follower through the operational rules.

The weight $y + 1$ has a different operational role from the weights generated during the aggregation phase. It does not represent additional informational support or new evidence acquired by the leader. Rather, it is a corrective priority weight chosen so that the message preserves the determinate label of the leader even when it is combined with an opposite unit-weight local state. In particular,

$C_{y+1} \star I_1 = C_y$ and $I_{y+1} \star C_1 = I_y$. Thus, in the opposite determinate case, the corrective update reproduces exactly the leader's weighted state. Qualitative alignment alone would require only a corrective weight of 2.

Lemma 3 (Corrective update). *Let $y \geq 1$. For every possible unit-weight local state of a follower:*

$$C_{y+1} \star C_1 = C_{y+2}, \qquad C_{y+1} \star I_1 = C_y, \qquad C_{y+1} \star U_1 = C_{y+1},$$

and

$$I_{y+1} \star I_1 = I_{y+2}, \qquad I_{y+1} \star C_1 = I_y, \qquad I_{y+1} \star U_1 = I_{y+1}.$$

Thus, once delivered and processed, γ_{y+1} ($\gamma \in \{C, I\}$) transforms any unit-weight follower state into a state with label γ.

Proof. The same-label equalities $C_{y+1} \star C_1 = C_{y+2}$ and $I_{y+1} \star I_1 = I_{y+2}$ follow from rules $[3_A]$ and $[3_B]$, respectively. The opposite-label equalities $C_{y+1} \star I_1 = C_y$ and $I_{y+1} \star C_1 = I_y$ follow from rules $[8_A]$ and $[8_B]$, because $y+1 > 1$. Finally, $C_{y+1} \star U_1 = C_{y+1}$ and $I_{y+1} \star U_1 = I_{y+1}$ follow from rules $[4_A]$ and $[4_B]$. Thus, in every possible case, the resulting qualitative label coincides with the label carried by the corrective message. □

During the retraction phase, the normalized state of the leader remains fixed, no new environmental measurement is introduced, and every selected follower is targeted by at most one corrective message. The message is used only to update the local state of that follower and does not modify the matrix or initiate another aggregation round. Qualitative alignment is guaranteed only for corrective messages that are successfully delivered and processed.

4.5 Termination and Qualitative Alignment

The following result establishes termination and qualitative alignment for the retraction phase under an explicit delivery assumption. The alignment is with the normalized collective assessment, independently of its external correctness. Thus, the result establishes an internal protocol property and should not be read as alignment with ground truth or as a fault-tolerance guarantee.

Theorem 4 (Termination and Qualitative Alignment). *Let P be a communication round involving follower agents $\{N_1, \ldots, N_n\}$ and a leader agent N_α. Suppose that the normalized state of the leader at time t_4 is $\gamma_y(N_\alpha)$, where $\gamma \in \{C, I\}$ and $y \geq 1$. For each follower N_i, let $s_i \in \{C_1, I_1, U_1\}$ be its local state at the beginning of the retraction phase t_5, and let $r_i \in \{C_1, I_1, U_1\}$ be the contribution recorded by the leader at time t_3. We assume that, whenever $r_i \neq U_1$, the follower retains its reported assessment until the beginning of the retraction phase, so that $s_i = r_i$.*

Assume that during time t_5:

1. *no new measurement or report is added to the completed round, and a follower's local state changes only when it processes its corrective message, if any;*

2. *the normalized state of the leader and the matrix $M(\Gamma_\alpha)$ remain unchanged;*
3. *every follower such that $r_i = U_1$ or $\mathrm{lab}(r_i)$ is the determinate label opposite to γ is sent exactly one corrective message γ_{y+1}, and that message is eventually delivered and processed by the follower;*
4. *no corrective message generates a further retraction during the same round.*

Then the retraction phase terminates after at most n corrective messages. In the resulting configuration, every follower carries the qualitative label γ.

If the successful-delivery-and-processing condition in item 3 *is dropped, the one-message-per-follower policy still guarantees that at most n corrective messages are generated, but completion of the corrective updates and qualitative alignment of followers whose messages are not delivered and processed are not guaranteed.*

Proof. Fix a follower N_i.

Suppose first that r_i already carries the label γ. Then the contribution from N_i was successfully received at time t_3, since a communication timeout would have produced the recorded contribution U_1. Hence $r_i = s_i$, and the local state of N_i already carries the label γ. Such a follower is not selected for correction.

Suppose next that r_i carries the determinate label opposite to γ. Again, the report was successfully received, so $r_i = s_i$. By assumption 3, the follower receives and processes exactly one corrective message γ_{y+1}. By Lemma 3, the resulting local state carries the label γ.

Finally, suppose that $r_i = U_1$. Whether this value results from a successfully transmitted undefined assessment or from a communication timeout, the actual local state s_i may be any of C_1, I_1, or U_1. By assumption 3, the follower receives and processes exactly one corrective message γ_{y+1}. By Lemma 3, the resulting local state carries the label γ.

Thus every follower carries the qualitative label γ after its possible corrective update. Each follower is targeted by at most one corrective message, and there are only n followers. Therefore, at most n corrective messages are generated. By assumption 3, every such message is eventually delivered and processed. By assumptions 1, 2, and 4, no new contribution can reopen the completed aggregation and no corrective message generates an additional retraction during the same round. Hence the retraction phase terminates after at most n corrective messages. □

5 Illustrative Scenario: Multi-Agent Coordination in Search and Rescue

In this section, we illustrate our framework in an AI-assisted wilderness Search and Rescue (SAR) system deployed in the Montuosa Forest, a fictional nature reserve defined by dense vegetation, rocky terrain, and multiple streams. The system is designed to support a reduced team of human rangers in managing search operations. In such high-stakes operations, human operators, autonomous systems (e.g., drones, ground robots), and stationary sensors must exchange

timely and reliable information to support coordinated decision-making and ensure mission effectiveness (see, e.g., [20, 18, 13] for real-world applications). In our scenario, during a family visit to the reserve, eight-year-old John becomes separated from his parents and cannot find his way back to the campground. After an unsuccessful initial search, the Montuosa SAR Command Center deploys a leader drone and ten ground robots to survey the surrounding forest sectors.

The system architecture is hierarchically structured into three functional levels:

1. **Human Command**: Human rangers receive processed data from the system and make high-level operational decisions based on the aggregated information.
2. **Leader Agent**: Acting as a coordination hub, the leader drone collects and processes input from the ground units, assigns search sectors, and applies the communication and computation rules described in Section 4.1.
3. **Follower Agents**: Ground robots perform area exploration using a Boustrophedon Coverage Algorithm combined with a Hierarchical Rechecking Strategy to enhance spatial coverage and search reliability [12, 8, 4].

Communication within the system follows a star-shaped topology from the follower agents to the leader. Each follower produces a unit-weight assessment C_1, I_1, or U_1 and transmits it to the leader; if an expected message is not received within the prescribed communication interval, the leader records U_1 for the corresponding follower. The leader then aggregates the recorded contributions according to the hierarchical protocol described in Section 4, constructs $M(\Gamma_\alpha)$, and adopts the corresponding normalized state.

We illustrate the search process in three steps in order to show how the operational rules handle determinate, undefined, and conflicting contributions during successive search rounds. As previously described, each step consists of surveying a specific area, which each follower agent is potentially capable of completing within a time frame of five minutes.[4] For illustration, we assume that the leader processes the recorded contributions in the order $N_1, \ldots, N_{10}$. Within each round, all agents evaluate the same proposition ϕ: "A trace attributable to John is present in the designated search area during the sensing interval."

Step 1. Uniform Negative Outcome In this first phase, all follower agents are assigned to survey a designated sector of the forest. After conducting their individual assessments, none of the agents find any indication or evidence related to the missing child. As a result, each follower agent assigns the label I to the proposition ϕ, signaling a negative outcome. They then transmit their evaluations to the leader agent:

$$I(N_1),\ I(N_2),\ \ldots,\ I(N_{10}).$$

[4] For the sake of simplicity, the example does not consider communication timeouts. Measurement failures, however, are included in order to illustrate the operational role of the label U.

Applying rule 0_A followed by nine applications of rule 3_B, the leader computes the final state as:

$$I_{10}(N_\alpha).$$

The weight 10 reflects complete consensus among the follower agents. Given that the result is both negative and consistent across the entire team, no conflicts arise, and no further corrections are necessary at this stage. The search operation continues in a new sector of the forest.

Step 2. Conflicting Input and Retraction Propagation In the second phase, a new forest sector is surveyed. Agent N_{10} detects what appears to be a clue and assigns the label C to proposition ϕ. The remaining agents are then directed toward the same location to perform their own measurements and transmit their evaluations to the leader. All other agents disagree with N_{10} and assign the label I, with the exception of N_8, who—due to a malfunction—fails to carry out the measurement and transmits the label U. The resulting configuration is:

$$I(N_1), \ldots, I(N_7), U(N_8), I(N_9), C(N_{10}).$$

The leader applies rule 0_A, followed by six applications of rule 3_B, one application of rule 6_B, one further application of rule 3_B, and finally rule 8_A. The resulting informational state is:

$$I_7(N_\alpha).$$

The weight 7 represents the residual negative support remaining after the positive contribution of N_{10} has been canceled against the accumulated negative contributions. The undefined contribution of N_8 does not affect this determinate balance. In this example, N_8 failed to produce a valid measurement, while N_{10} produced the only positive assessment.

Following the rules for *Propagation of Retractions*, the normalized state of the leader remains $I_7(N_\alpha)$. Since N_{10} produced the discrepant report C_1 and N_8 produced the undefined report U_1, rule R_2 generates one corrective message for each agent:

$$\mathrm{msg}^{t_5}_{\alpha\to 10}(I_8) \qquad \text{and} \qquad \mathrm{msg}^{t_5}_{\alpha\to 8}(I_8).$$

The message weight is $8 = 7 + 1$ and does not modify the state of the leader. It is processed only by the receiving follower. By rules $[8_B]$ and $[4_B]$, respectively,

$$I_8 \star C_1 = I_7 \qquad \text{and} \qquad I_8 \star U_1 = I_8.$$

Thus, both followers assume the qualitative label I. In this example, the U_1 contribution of N_8 results from its failure to complete a valid measurement. The positive contribution of N_{10} is instead discrepant with the normalized negative assessment and is therefore selected for corrective feedback. This discrepancy alone does not establish that N_{10} is faulty.

The correction phase has two effects: (i) the normalized state of the leader remains stable; (ii) the selected followers are qualitatively aligned with the normalized assessment of the leader; any new measurement, if required, belongs to a subsequent round.

Final Step. Positive Consensus and Rescue In a subsequent survey of a new forest sector, agent N_2 identifies a potential clue during its measurement and assigns the label C to proposition ϕ. As in the previous case, all other agents are redirected to the same location to perform their own evaluations. This time, all agents independently obtain a positive assessment, each assigning the label C to ϕ and transmitting the result to the leader agent:

$$C(N_1),\, C(N_2),\, \ldots,\, C(N_{10}).$$

Applying rule 0_A followed by nine applications of rule 3_A, the leader agent computes the final state $C_{10}(N_\alpha)$. The leader then transmits the normalized positive assessment to the human command level, which can use this information to prioritize intervention in the corresponding search sector.

This example illustrates how the proposed framework processes determinate, undefined, and conflicting contributions, preserves a normalized collective assessment, and provides post-aggregation feedback to followers associated with undefined or discrepant reports.

6 Conclusion and Future Work

We introduced an operational framework for information aggregation in multi-agent systems based on four weighted informational labels: C and I for positive and negative determinate assessments, U for the absence of a usable determinate assessment, and B for unresolved conflict. The operational rules support incremental aggregation and guarantee that, for every fixed finite collection of contributions, the final qualitative label is independent of the processing order. The complete weighted state is also order-independent for C-, I-, and U-outcomes, whereas a sequential computation ending in B may retain an order-dependent provisional weight. To remove this residual dependence, we introduced an order-independent matrix representation of the completed round. The matrix agrees with sequential aggregation on the qualitative result and, for C, I, and U, also on the weight; in the conflict case, it assigns the canonical weight used to define the normalized final state. The framework was then specialized to a hierarchical leader–follower architecture equipped with a post-aggregation feedback mechanism. Under the stated delivery and stability assumptions, the corrective phase terminates after finitely many messages and qualitatively aligns the selected followers with the normalized collective assessment. The search and rescue scenario illustrates these mechanisms in a concrete multi-agent setting without being intended as an empirical performance evaluation. This alignment is likewise internal to the aggregation protocol: qualitative alignment with the normalized collective assessment is not a claim of convergence to external truth.

Several directions remain open. First, the four informational labels suggest a possible connection with non-classical logical semantics. At the qualitative level, one may consider the projection $q(C) = 1$, $q(I) = 0$, $q(U) = q(B) = \mathsf{u}$, which collapses the distinction between undefined information and conflict. This projection does not by itself define a 3-valued logic: the aggregation operation is

not a logical connective, and establishing connections with 3-valued systems [14, 7, 17, 2] requires a separate analysis.

Further work may extend the framework to richer temporal models in which later reports revise or supersede earlier contributions, and to settings incorporating probabilistic information, sensor reliability, or independently assigned trust values [1]. Another theoretical direction is to characterize the algebraic structure induced by $\star$ and its relation to matrix normalization. The operation is commutative but not associative on weighted informational states: $(U_1 \star I_1) \star C_1 = B_1$, whereas $U_1 \star (I_1 \star C_1) = B_2$. It would be useful to study suitable quotients, abstractions, or normalizations, and to determine which quantitative information they preserve. The invariants σ and κ, together with the matrix normalization, provide a starting point for this analysis. Such extensions could also be studied through probabilistic verification tools such as PRISM [15] or VESTA [21]. Finally, an implementation in rewriting logic, for instance using Maude [9], would provide a natural environment for representing the operational rules and studying scalability, communication failures, and execution under more realistic conditions.

On the practical side, we plan to investigate the integration of our framework into cloud storage and service orchestration scenarios, particularly in Kubernetes-based environments [23]. Kubernetes coordinates multiple components, including nodes, pods, services, and controllers, whose local observations may be incomplete, contradictory, or asynchronously updated due to network delays or component failures. We aim to explore how our framework can reconcile such information, reason about inconsistencies between the desired and actual cluster states, and support the monitoring and coordination of shared state across the cluster.

References

1. Alessandro Aldini, Gianluca Curzi, Pierluigi Graziani, and Mirko Tagliaferri. A probabilistic modal logic for context-aware trust based on evidence. *International Journal of Approximate Reasoning*, 169:109167, 2024.
2. Alessandro Aldini, Pierluigi Graziani, Claudio Antares Mezzina, and Gandolfo Vergottini. A logical 3-valued semantics for nondeterministic choice. In *Proceedings 21st International Symposium on Logical and Semantic Frameworks with Applications (LSFA 2026)*, volume 449. Electronic Proceedings in Theoretical Computer Science, 2026.
3. Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.
4. Ruofei Bai, Ronghao Zheng, Yang Xu, Meiqin Liu, and Senlin Zhang. Hierarchical multi-robot strategies synthesis and optimization under individual and collaborative temporal logic specifications. *Robotics and Autonomous Systems*, 153:104085, 2022.
5. Nuel D. Belnap. How a computer should think. In G. Ryle, editor, *Contemporary Aspects of Philosophy*, pages 30–55. Oriel Press, Stocksfield, 1977.

6. Nuel D. Belnap. A useful four-valued logic. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*. D. Reidel, 1977.
7. Dmitri A Bochvar and Merrie Bergmann. On a three-valued logical calculus and its application to the analysis of the paradoxes of the classical extended functional calculus. *History and Philosophy of Logic*, 2(1-2):87–112, 1981.
8. Howie Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9:247–253, 12 2000.
9. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn L. Talcott, editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.
10. Emilio Cruciani, Andrea Clementi, Francesco Scornavacca, and Francesco Pasquale. Phase transitions in binary consensus protocols for multi-agent systems. *Distributed Computing*, 34(3):215–235, 2021.
11. Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, 1985.
12. Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.
13. Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: the robot world cup initiative. *Proceedings of the International Conference on Autonomous Agents*, 04 1998.
14. Stephen C. Kleene. *Introduction to Metamathematics*. North-Holland, 1952.
15. Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: probabilistic model checking for performance and reliability analysis. *SIGMETRICS Perform. Evaluation Rev.*, 36(4):40–45, 2009.
16. Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
17. John McCarthy. A basis for a mathematical theory of computation. In *Computer Programming and Formal Systems*, pages 33–70. North-Holland, 1963.
18. Robin R. Murphy. Search and rescue robotics. In *Springer Handbook of Robotics*, pages 1151–1173. Springer, 2008.
19. Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
20. Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8:191617–191643, 2020.
21. Koushik Sen, Mahesh Viswanathan, and Gul A. Agha. VESTA: A statistical model-checker and analyzer for probabilistic systems. In *Second International Conference on the Quantitative Evaluation of Systems (QEST 2005)*, pages 251–252. IEEE Computer Society, 2005.
22. Philippe Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447–458, 1990.
23. The Kubernetes Authors. Kubernetes Documentation. https://kubernetes.io/docs/, 2026. [Online; accessed 30-Oct-2026].