Multi-Objective Multi-Agent Planning for Discovering and Tracking Multiple Mobile Objects

Hoa Van Nguyen, Ba-Ngu Vo, Ba-Tuong Vo, Hamid Rezatofighi and Damith C. Ranasinghe

***Abstract*—We consider the online planning problem for a team of agents to discover and track an unknown and time-varying number of moving objects from onboard sensor measurements with uncertain measurement-object origins. Since the onboard sensors have limited field-of-views, the usual planning strategy based solely on either tracking detected objects or discovering unseen objects is inadequate. To address this, we formulate a new information-based multi-objective multi-agent control problem, cast as a partially observable Markov decision process (POMDP). The resulting multi-agent planning problem is exponentially complex due to the unknown data association between objects and multi-sensor measurements; hence, computing an optimal control action is intractable. We prove that the proposed multi-objective value function is a monotone submodular set function, which admits low-cost suboptimal solutions via greedy search with a tight optimality bound. The resulting planning algorithm has a *linear* complexity in the number of objects and measurements across the sensors, and *quadratic* in the number of agents. We demonstrate the proposed solution via a series of numerical experiments with a real-world dataset.**

***Index Terms*—Stochastic control, path planning, multi-agent control, MPOMDP, multi-object tracking.**

I. Introduction

Recent advancements in robotics have inspired applications that use low-cost mobile sensors (*e.g.*, drones), ranging from vision-based surveillance, threat detection via source localisation, search and rescue, to wildlife monitoring [1]. At the heart of these applications is Multi-Object Tracking (MOT)—the task of estimating an unknown and time-varying number of objects and their trajectories from sensor measurements with unknown *data association* (*i.e.*, unknown measurement-to-object origin). Further, MOT is fundamental for autonomous operation as it provides awareness of the dynamic environment in which the agents operate. Although a single agent can be tasked with MOT, such a system is limited by observability, computing resources, and energy. Using multi-agents alleviates these problems, improves synergy, and affords robustness to failures. Realising this potential requires the multi-agents to collaborate and operate autonomously.

In this work, we consider the challenging problem of coordinating multiple limited field-of-view (FoV) agents to *simultaneously* seek *undetected objects* and track *detected objects* (see Fig. 1). Tracking involves estimating the trajectories of the objects and maintaining their provisional identities or labels [2]. Trajectories are important for capturing the behaviour of the objects while their labels provide the means for distinguishing individual trajectories and for human/machine users to communicate information on the relevant trajectories. A single agent with an on-board sensor (*e.g.*, a camera, a radar) invariably has a limited FoV [3], and hence, only observes part of the scene at any given time. As a result, a team of multiple agents is often implemented to improve coverage of the surveillance area [4]–[9]. However, MOT with multiple limited-FoV sensors still encountered several challenges, such as occlusions, missed detections, false alarms, identity switches and track fragmentation [10], [11].

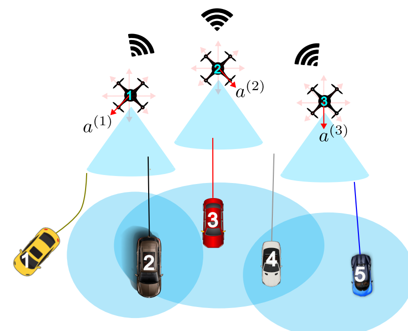

Fig. 1. An unmanned aerial vehicles (UAV) team tracking multiple vehicles with limited FoV sensors and unknown measurements-to-objects associations.

Discovering undetected objects and tracking detected objects are two competing objectives due to the limited FoVs of the sensors and the random appearance/disappearance of objects. On one hand, following only detected objects to track them accurately means that many undetected objects could be missed. On the other hand, leaving detected objects to explore unseen regions for undetected objects will lead to track loss. Thus, the problem of seeking *undetected objects* and tracking *detected objects* is a multi-objective optimisation problem.

Even for standard state-space models, where the system state is a finite-dimensional vector, multi-agent planning with multiple competing objectives is challenging due to complex interactions between agents resulting in combinatorial optimisation problems [12]. In MOT, where the system state (and measurement) is a set of vectors, the problem is further complicated due to: *i)* the unknown and time-varying number of trajectories; *ii)* missing and false detections; and *iii)* unknown data association (measurement-to-object origins) [13]. Most critically for real-world applications, multi-agent control actions must be computed online and in a timely manner.

Acknowledgement: This work is supported by the Australian Research Council under Projects LP160101177, LP200301507, and FT210100506.

Hoa Van Nguyen, Ba-Tuong Vo and Ba-Ngu Vo are with the Department of Electrical and Computer Engineering, Curtin University, Bentley, WA 6102, Australia (e-mail: hoa.v.nguyen,ba-tuong.vo,ba-ngu.vo@curtin.edu.au).

Hamid Rezatofighi is with the Department of Data Science & AI, Monash University, Clayton VIC 3800, Australia (e-mail: hamid.rezatofighi@monash.edu).

Damith C. Ranasinghe is with the School of Computer Science, The University of Adelaide, SA 5005, Australia (e-mail: damith.ranasinghe@adelaide.edu.au).

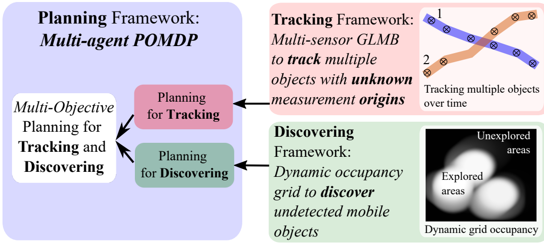

Fig. 2. A schematic of the proposed planning approach.

Model predictive control (MPC) is an effective approach to stochastic control and is widely used [14], compared to meta-heuristics (or bio-inspired) techniques such as genetic algorithms and particle swarm optimisation [15], which are expensive for real-time applications in dynamic environments such as MOT. The MPC problem can be cast as a partially observed Markov decision process (POMDP), which has been gaining significant interest as a real-time planning approach [16]. The cooperation problem amongst agents can be formulated as a decentralised POMDP (Dec-POMDP), whose exact solutions are NEXP-hard [17]. Moreover, for multi-agent Dec-POMDP, the action and observation space grows exponentially with the number of agents [17], and hence unsuitable for real-time applications. This intractability arises because the distributed agents do not necessarily have the same formation states, and computing the optimal action requires accounting for all possible observation histories and action sequences of the agents. In contrast, all agents in a centralised POMDP have the *same* information state (from the central node), which drastically reduces the complexity of action selection. Therefore, centralised Multi-agent POMDP [18] offers a more tractable alternative for coordinating multiple agents [19], [20], and is adopted in our work.

Computing optimal actions for MOT in a POMDP requires a suitable framework that provides a multi-object density for the information state. Amongst various MOT algorithms, we adopt the labelled random finite set (RFS) framework because it provides a multi-object filtering density that contains all information about the current set of trajectories and accommodates its time-varying nature. Single-agent planning with RFS filters has been studied with an unlimited FoV [21], [22] and extended to multi-agent planning using distributed fusion [20]. However, these multi-agent planning methods are only suitable when the agents have an unlimited detection range [10]. A multi-agent POMDP with an RFS filter was proposed in [19] for searching and localisation, but not tracking.

To achieve competing discovery and tracking objectives, we propose a POMDP with a multi-objective value function consisting of information gains for both detected and undetected objects, as illustrated in Fig. 2. A simple solution for competing objective functions is to weigh and add them together. However, meaningful weighting parameters are difficult to determine. A multi-objective optimisation approach naturally provides a meaningful trade-off between competing objectives. To the best of our knowledge, multi-agent planning for MOT with a multi-objective value function has not been investigated. The first multi-objective POMDP with a localisation RFS filter was proposed in [23] for a sensor selection problem.

For MOT and the information state, we use the multi-sensor Generalised Labelled Multi-Bernoulli (MS-GLMB) filter [24], which exhibits *linear* computational complexity in the total number of measurements across the sensors. For path planning, we use the centralised *Multi-Agent POMDP* (MPOMDP) framework [18] with the GLMB filter. Further, for the value function, we use the differential entropy [25, pp.243] of the Labelled Multi-Bernoulli (LMB) density [26] that matches the first moment of the information state. In particular, we derive an analytic expression for this differential entropy, which can be computed with *linear* complexity in the number of hypothesised objects, and show that the resulting multi-objective value function is monotone submodular. Consequently, this enables us to exploit low-cost greedy search to solve the optimal control problem with tight optimality bound. In summary, the main contributions are:

i) A multi-objective POMDP formulation for multi-agent planning to search and track multiple objects with unknown object-to-measurement origins.

ii) The concept of differential entropy and mutual information for RFS, and an analytic expression for the differential entropy of LMBs with *linear* complexity in the number of objects.

iii) An efficient multi-agent path planning algorithm with *linear* complexity in the number of objects and quadratic in the number of agents, using differential-entropy-based and occupancy-grid-based value functions.

The proposed method is evaluated using the **CRAWDAD taxi dataset** [27] on multi-UAVs control for searching and tracking an unknown and time-varying number of taxis in downtown San Francisco. For performance benchmarking, our preliminary work in [28] is used as the ideal or best-case scenario where data association is known.

The remainder of this paper is organised as follows. Section II provides relevant background for our MOT and planning framework. Section III presents our proposed multi-objective value function for the control of the multi-agent to simultaneously track and discover objects. Section IV evaluates our proposed method on a real-world dataset. Section V summarises our contributions and discusses future directions.

II. Background

This section presents the problem statement and necessary background on MPOMDP and MS-GLMB filtering. We use notation in accordance with [29]. Lowercase letters (*e.g.*, $x, \boldsymbol{x}$) denote vectors, uppercase letters (*e.g.*, $X, \boldsymbol{X}$) denote finite sets, while spaces are denoted by blackboard uppercase letters (*e.g.*, $\mathbb{X}, \mathbb{L}$). Vectors augmented with labels, finite sets of labelled vectors, and the corresponding spaces are bolded (*e.g.*, $\boldsymbol{x}, \boldsymbol{X}$, $\boldsymbol{\mathbb{X}}, \boldsymbol{\mathbb{U}}$) to distinguish them from unlabelled ones. For a given set S, $|S|$, $1_S(\cdot)$, $\delta_S[\cdot]$ denote, respectively, the cardinality, indicator function, and (generalised) Kronecker delta function, of S ($\delta_S[X] = 1$, if $X = S$, and zero otherwise). For a function f, its multi-object exponential f^X is defined as $\prod_{x\in X} f(x)$, with $f^{\emptyset} = 0$. The inner product $\int f(x)g(x)dx$ is written as $\langle f, g\rangle$, while the superscript † denotes the transpose of a vector/matrix. For compactness, the subscript for time k is

TABLE I
LIST OF SYMBOLS

Symbols	Description
$\mathcal{N}$	finite set of agent labels
$\mathcal{G}$	Gaussian distribution
H	look-ahead horizon
$\mathbb{X}$	unlabelled single-object state space
$\mathbb{L}$	label space
$\mathbf{X} \triangleq \mathbb{X} \times \mathbb{L}$	labelled single-object state space
$\boldsymbol{\mathcal{X}}$	partition matroid of $\mathbf{X}$
$\boldsymbol{X} \in \boldsymbol{\mathcal{X}}$	labelled multi-object state
$\mathbb{U}^{(n)}$	state space of agent n
$u^{(n)} \in \mathbb{U}^{(n)}$	state of agent n
$\mathbb{Z}^{(n)}$	observation space of agent n
$\mathbf{Z} \triangleq \uplus_{n\in\mathcal{N}}(\mathbb{Z}^{(n)} \times \{n\})$	common observation space
$\boldsymbol{\mathcal{Z}}$	class of finite subsets of $\mathbf{Z}$
$\boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$	multi-agent multi-object observation
$\mathbb{A}^{(n)}$	control action space of agent n
$\mathbb{A} = \mathbb{A}^{(1)} \times ... \times \mathbb{A}^{(\lvert\mathcal{N}\rvert)}$	multi-agent action space
$\mathbf{A} \triangleq \uplus_{n\in\mathcal{N}}(\mathbb{A}^{(n)} \times \{n\})$	common action space
$\boldsymbol{\mathcal{A}}$	partition matroid of $\mathbf{A}$
$a = (a^{(1)}, ..., a^{(\lvert\mathcal{N}\rvert)})$	multi-agent action
$\boldsymbol{\varrho}$	immediate reward function
h	differential entropy function
V_1	tracking value function
V_2	discovery value function
V_{mo}	multi-objective value function
$\kappa^{(i)}$	occupancy grid cell i
$O^{(i)}$	Bernoulli random variable of cell i
$Y^{(i)}$	binary observation of cell i

omitted, and subscripts for times $k-1$ and $k+1$ are abbreviated '$-$' and '$+$'. A list of symbols is provided in Table I.

A. Problem statement

Consider a team of agents, equipped with limited-FoV sensors that record detections with unknown *data association* (measurement-to-object origins), monitoring an unknown and time-varying number of objects. The agents can self-localise and communicate (*e.g.*, sending observations) to a central node that determines/issues control actions [19]. For each agent $n\in\mathcal{N} = \{1,\ldots,|\mathcal{N}|\}$, its state space[1], (discrete) control action space, and observation space, are denoted respectively by $\mathbb{U}^{(n)}$, $\mathbb{A}^{(n)}$, and $\mathbb{Z}^{(n)}$. We define the *common observation space* for all agents as $\mathbf{Z} \triangleq \uplus_{n\in\mathcal{N}}\big(\mathbb{Z}^{(n)} \times \{n\}\big)$.

Each object is described by a labelled state $\boldsymbol{x} = (x,\ell) \in \mathbf{X} \triangleq \mathbb{X} \times \mathbb{L}$, consisting of a time-varying *attribute* $x \in \mathbb{X}$, and a time-invariant discrete *label* $\ell \in \mathbb{L}$. To capture the time-varying number of trajectories, the *multi-object state* at any given time is represented by the set $\boldsymbol{X}$ of distinctly labelled states of the objects [29], [30]. Thus, the *multi-object state space* $\boldsymbol{\mathcal{X}}$ is the *partition matroid* of $\mathbf{X}$. A partition matroid of a space $\mathbf{S} = \mathbb{S} \times \mathbb{I}$, with $\mathbb{I}$ a discrete set, is the class of subsets of $\mathbf{S}$, defined by [31]:

$$\boldsymbol{\mathcal{S}} \triangleq \{\boldsymbol{S} \subseteq \mathbf{S} : |\boldsymbol{S} \cap (\mathbb{S} \times \{i\})| \leq 1,\ \forall i \in \mathbb{I}\}.$$

Each member of the partition matroid $\boldsymbol{\mathcal{S}}$ has distinct indices.

[1]The measurement errors of agents' internal actuator-sensor are assumed negligible, thus the agent state is known.

Our goal is to coordinate the team of agents to search and track a time-varying and unknown number of moving objects. The main challenges are: *i)* misdetection of objects due to the limited FoVs; *ii)* lack of prior information about the changing number of objects and their locations; *iii)* unknown data association; and *iv)* the computational resources required to determine multi-agent control actions in a timely manner.

B. MPOMDP for Multi-Object Tracking

A random finite set (RFS) model [32], [33] succinctly captures the random nature of the multi-object states and offers a suitable concept of multi-object density for the POMDP's *information state*, encapsulating all current knowledge about the system based on the observation history. Specifically, *an RFS of a space* $\mathbb{X}$ *is a random variable taking values in the class of finite subsets of* $\mathbb{X}$, and is commonly characterised by Mahler's *multi-object density* [32] (see also subsection III-A). Since the elements of a multi-object state are distinctly labelled, we model the multi-object state as a *labelled RFS*–an RFS of $\mathbf{X}$ such that instantiations have distinct labels [29], i.e., a random variable taking values in the partition matroid $\boldsymbol{\mathcal{X}}$.

Conceptually, an MPOMDP is similar to a single-agent POMDP, but incorporates joint models of all agents and communications for sharing observations and/or information [18]. This work explores a centralised approach[2] where agents send observations to a central node responsible for coordinating the team via the control actions. The multi-object MPOMDP considered in this work can be characterised by the tuple $(\mathcal{N}, H, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Z}}, \mathbb{A}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{\varrho})$, where

- $\mathcal{N}$ is the finite set of agent labels;
- H is the look-ahead time horizon;
- $\boldsymbol{\mathcal{X}}$ is the multi-object state space (partition matroid of $\mathbf{X}$);
- $\boldsymbol{\mathcal{Z}}$ is the multi-object observation space (class of finite subsets of $\mathbf{Z}$);
- $\mathbb{A} = \mathbb{A}^{(1)} \times ... \times \mathbb{A}^{(|\mathcal{N}|)}$ is multi-agent action space;
- $\boldsymbol{f}(\boldsymbol{X}|\boldsymbol{X}_-, a_-)$ is the transition density for $\boldsymbol{X}, \boldsymbol{X}_- \in \boldsymbol{\mathcal{X}}$, and action $a_- \in \mathbb{A}$ has been taken;
- $\boldsymbol{g}(\boldsymbol{Z}|\boldsymbol{X}, a_-)$ is the likelihood of observation $\boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$ given $\boldsymbol{X} \in \boldsymbol{\mathcal{X}}$, and action $a_- \in \mathbb{A}$ has been taken;
- $\boldsymbol{\varrho}(\boldsymbol{Z}, \boldsymbol{X}, a_-)$ is the immediate reward for observing $\boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$ from $\boldsymbol{X} \in \boldsymbol{\mathcal{X}}$, after action $a_- \in \mathbb{A}$ has been taken.

The agents are coordinated at each time k via the multi-agent action $a = (a^{(1)}, ..., a^{(|\mathcal{N}|)}) \in \mathbb{A}$, which prescribes a state for each agent n at time $k+1$, where it collects the multi-object observation $Z_+^{(n)} \subset \mathbb{Z}^{(n)}$ from the multi-object state $\boldsymbol{X}_+$ that evolved from $\boldsymbol{X}$ according to the multi-object transition density $\boldsymbol{f}_+(\boldsymbol{X}_+|\boldsymbol{X}, a)$. The multi-agent multi-object observation $\boldsymbol{Z}_+ \in \boldsymbol{\mathcal{Z}}$ is given by $\uplus_{n\in\mathcal{N}}\big(Z_+^{(n)} \times \{n\}\big)$, or equivalently the $|\mathcal{N}|$-tuple $(Z_+^{(1)} \times \{1\}, \ldots, Z_+^{(|\mathcal{N}|)} \times \{|\mathcal{N}|\})$, and has likelihood $\boldsymbol{g}_+(\boldsymbol{Z}_+|\boldsymbol{X}_+, a)$. Note that, by convention $Z_+^{(n)} \times \{n\} = \emptyset$ when agent n is inactive (broken down), which is different from $Z_+^{(n)} \times \{n\} = \emptyset \times \{n\}$ when n is active but collects the empty observation (e.g., because there is nothing in its FoV). The specifics of $\boldsymbol{f}_+$ and $\boldsymbol{g}_+$ are detailed in Subsection (II-C).

[2]The communication traffic between agents and a central node is minimal if employing detection-based observations.

The *information state* of the MPOMDP at time $j > 1$ is the (labelled) *multi-object filtering density* $\boldsymbol{\pi}_{a_{j-1},j}(\cdot|\boldsymbol{Z}_j)$, which contains all information about the multi-object state given the observation history [29]. For notational convenience, we indicate dependence only on the latest control action and observation. Starting with the initial prior $\boldsymbol{\pi}_{a_0,1}$, the information state $\boldsymbol{\pi}_{a_{j-1},j}(\cdot|\boldsymbol{Z}_j)$ is propagated via Bayes recursion:

$$\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}_j) = \int \boldsymbol{f}_j(\boldsymbol{X}_j|\boldsymbol{X}, a_{j-1})\boldsymbol{\pi}_{a_{j-2},j-1}(\boldsymbol{X})\,\delta\boldsymbol{X}, \quad (1)$$

$$\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}_j|\boldsymbol{Z}_j) = \frac{\boldsymbol{g}_j(\boldsymbol{Z}_j|\boldsymbol{X}_j, a_{j-1})\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}_j)}{\boldsymbol{g}_j(\boldsymbol{Z}_j|a_{j-1})}, \quad (2)$$

where $\boldsymbol{g}_j(\boldsymbol{Z}_j|a_{j-1}) = \int \boldsymbol{g}_j(\boldsymbol{Z}_j|\boldsymbol{X}_j, a_{j-1})\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}_j)\delta\boldsymbol{X}_j$ is the *predictive likelihood* under action a_{j-1}, and the integral is Mahler's set integral (see also Subsection III-A for details).

A POMDP seeks the control action sequence $a_{k:k+H-1}$ that maximises a *value function* $V\big(a_{k:k+H-1}\big)$ constructed from the information states and immediate rewards over horizon H, e.g., the expected sum of immediate rewards

$$V\big(a_{k:k+H-1}\big) = \quad (3)$$
$$\sum_{j=k+1}^{k+H} \iint \boldsymbol{\varrho}_j(\boldsymbol{Z}, \boldsymbol{X}, a_{j-1})\,\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}|\boldsymbol{Z})\boldsymbol{g}_j(\boldsymbol{Z}|a_{j-1})\delta\boldsymbol{X}\delta\boldsymbol{Z}.$$

In general, the value function (3) cannot be evaluated analytically, while numerical evaluations are intractable [34]. Especially for the general case, the control action a_j can be changed at each time step. Consequently, the action space grows exponentially over the horizon, leading to the intractability of computing the value function (3). An approximate strategy is to assume $a_j = a$ for $j = k : k+H-1$, and determine the action a that optimises the resulting value function, but only apply this action for $M < H$ time steps. The process is then repeated at time $M+1$. For single-angle planning for tracking problems, this approximate search strategy has shown to be effective in solving practical problems [1], [35]. However, for multi-object multi-agent problems, this approximation strategy is still too computationally intensive [36].

In this work, we consider a computationally tractable value function based on the notion of *Predicted Ideal Measurement Set* (PIMS) [13]. Using the information state $\boldsymbol{\pi}_{a_{k-1},k}(\cdot|\boldsymbol{Z}_k)$ at time k, we first compute the multi-object prediction density

$$\widehat{\boldsymbol{\pi}}_{a,j}(\boldsymbol{X}_j) = \int \prod_{i=k+1}^{j} \boldsymbol{f}_i(\boldsymbol{X}_i|\boldsymbol{X}_{i-1}, a)\boldsymbol{\pi}_{a_{k-1},k}(\boldsymbol{X}_k)\,\delta\boldsymbol{X}_{k:j-1},$$

from which the *predicted multi-object state* $\widehat{\boldsymbol{X}}_{a,j}$ is determined via some multi-object estimator. The PIMS $\widehat{\boldsymbol{Z}}_{a,j}$ is the multi-object observation of $\widehat{\boldsymbol{X}}_{a,j}$ without noise nor false positives/negatives and known (object-to-measurement) origins, after action a has been applied at times k to $k+j-1$ [13]. The PIMS-based value function is defined as

$$V(a) = \sum_{j=k+1}^{k+H} \int \boldsymbol{\varrho}_j(\widehat{\boldsymbol{Z}}_{a,j}, \boldsymbol{X}_j, a)T(\boldsymbol{\pi}_{a,j}(\boldsymbol{X}_j|\widehat{\boldsymbol{Z}}_{a,j}))\delta\boldsymbol{X}_j, \quad (4)$$

where T is some transformation of the information state, e.g., a distribution constructed from its moments. The ideal measurements are the most informative measurements possible. Real sensor observations (with noise, false positives/negatives and unknown origins) "close" to the ideal measurements are more informative than those further away. Hence optimising the PIMS value function would yield actions that result in the agents collecting informative observations.

Remark 1: The exclusion of noise and false positives/negatives (present in real sensors) from the ideal measurements does not mean the PIMS approach neglects the limitations of real sensors. Rather, the rationale is to reward actions that result in (real) observations with less noise and false positives/negatives.

C. MS-GLMB Filtering

1) Multi-Object Dynamic Model: The multi-object transition density $\boldsymbol{f}_+$ captures the motions and births/deaths of objects. At time k, an object with state $\boldsymbol{x} = (x, \ell) \in \boldsymbol{X}$ either survives to the next time with probability $r_{S,+}(\boldsymbol{x})$ or dies with probability $1 - r_{S,+}(\boldsymbol{x})$ [29]. Conditional on survival it takes on the state $\boldsymbol{x}_+ = (x_+, \ell_+)$ according to the transition density $f_+(x_+|x, \ell)\delta_\ell[\ell_+]$, where $\delta_\ell[\ell_+]$ ensures the label remains the same. In addition, an object with state $\boldsymbol{x}_+ = (x_+, \ell_+)$ is born at time $k+1$ with probability $r_{B,+}(\ell_+)$, and its unlabelled state x_+ is distributed according to the probability density $p_{B,+}(\cdot, \ell)$. It is standard practice to assume that, conditional on the current multi-object state, objects are born or displaced at the next time, independently of one another [13]. For the search and track problem considered here, the objects themselves are not influenced by the multi-agent actions (i.e., the objects are not intelligent), and hence the multi-object transition density $\boldsymbol{f}_+(\boldsymbol{X}_+|\boldsymbol{X})$ is independent of a. This also implies the independence of agent n's observation from other agents' actions. An explicit expression for $\boldsymbol{f}_+(\boldsymbol{X}_+|\boldsymbol{X})$ based on the described model can be found in [29], though this is not required in this work.

2) Multi-Object Observation Model: The multi-object observation likelihood $\boldsymbol{g}$ captures detections/misdections, false alarms (or clutter), and data association uncertainty. Given a multi-object state $\boldsymbol{X}$, each $\boldsymbol{x} \in \boldsymbol{X}$ has probability $P_D^{(n)}(\boldsymbol{x}, u^{(n)})$ of being detected by agent n with state $u^{(n)}$, and generates an observation $\boldsymbol{z} = (z, n') \in \mathbb{Z}$ with likelihood [36]:

$$g^{(n)}(\boldsymbol{z}|\boldsymbol{x}, u^{(n)}) = \delta_n[n']g^{(n)}(z|\boldsymbol{x}, u^{(n)}),$$

or missed with probability $1 - P_D^{(n)}(\boldsymbol{x}, u^{(n)})$. The term $\delta_n[n']$ ensures that detections recorded by agent n are tagged with label n. The observation set collected by this agent is formed by the superposition of the (object-originated) detections and Poisson clutter with intensity

$$\kappa^{(n)}(\boldsymbol{z}|u^{(n)}) = \delta_n[n']\kappa^{(n)}(z|u^{(n)}).$$

It is standard practice to assume that conditional on $\boldsymbol{X}$, detections are independent of each other and clutter [32], and that the observations obtained by individual agents are independent [37].

Remark 2: Since the multi-agent state is determined by the control action $a = (a^{(1)}, ..., a^{(|\mathcal{N}|)})$, we indicate this dependence by writing $P_D^{(n)}(\boldsymbol{x}, a^{(n)}) \triangleq P_D^{(n)}(\boldsymbol{x}, u^{(n)})$,

$\kappa^{(n)}(\boldsymbol{z}|a^{(n)}) \triangleq \kappa^{(n)}(\boldsymbol{z}|u^{(n)})$, and $g^{(n)}(\boldsymbol{z}|\boldsymbol{x},a^{(n)}) \triangleq g^{(n)}(\boldsymbol{z}|\boldsymbol{x},u^{(n)})$, unless otherwise stated.

Due to the unknown data association, it is necessary to consider all multi-agent association maps, i.e., combinations of measurement-to-object origins. More concisely, a multi-agent *association map* γ is an $|\mathcal{N}|$-tuple of *positive 1-1 maps*[3]

$$\gamma^{(n)}: \mathbb{L} \to \{-1{:}|\boldsymbol{Z}^{(n)}|\}, n = 1, ..., |\mathcal{N}|,$$

where $\gamma^{(1)}(\ell) = ... = \gamma^{(|\mathcal{N}|)}(\ell) = -1$ if label ℓ is dead/unborn, $\gamma^{(n)}(\ell) = 0$ if object ℓ is not detected by agent n, and $\gamma^{(n)}(\ell) > 0$ if object ℓ generates observation $\boldsymbol{z}_{\gamma^{(n)}(\ell)}$ at agent n [24]. The positive 1-1 property ensures each observation from agent n originates from at most one object. The space of all such multi-agent associations is denoted as Γ, and the set $\mathcal{L}(\gamma) \triangleq \{\ell \in \mathbb{L} : \gamma^{(1)}(\ell), ..., \gamma^{(|\mathcal{N}|)}(\ell) \geq 0\}$ is called the *live labels* of γ. It is assumed that conditional on the multi-object state, the measurements from individual agents are independent (i.e., there is no interference among agents in the observation process) [37]. An explicit expression for $\boldsymbol{g}\left(\boldsymbol{Z}|\boldsymbol{X},a\right)$ based on the described model can be found in [24], though this expression is not needed in this work.

3) MS-GLMB Filter: The Bayes recursion (1)-(2) admits an analytical information state (multi-object filtering density) in the form of a GLMB [29]

$$\boldsymbol{\pi}\left(\boldsymbol{X}\right) = \sum_{I,\xi} w^{(I,\xi)} \delta_I[\mathcal{L}\left(\boldsymbol{X}\right)] \left[p^{(\xi)}\right]^{\boldsymbol{X}}, \quad (5)$$

where: $\mathcal{L}\left(\boldsymbol{X}\right)$ denotes the labels of $\boldsymbol{X}$; each I is a finite subset of $\mathbb{L}$; each ξ is a history $\gamma_{1:k}$ of multi-agent association maps up to the current time; each $w^{(I,\xi)}$ is non-negative such that $\sum_{I,\xi} w^{(I,\xi)} = 1$; and each $p^{(\xi)}\left(\cdot,\ell\right)$ is a probability density on the attribute space $\mathbb{X}$. Note that the traditional GLMB form involves the distinct label indicator $\delta_{|\boldsymbol{X}|}\left[|\mathcal{L}\left(\boldsymbol{X}\right)|\right]$, which is not needed here since the multi-object state space is the partition matroid $\boldsymbol{\mathcal{X}}$. For convenience, we abbreviate the GLMB in (5) by the set of components, i.e., $\boldsymbol{\pi} \triangleq \left\{\left(w^{(I,\xi)}, p^{(\xi)}\right)\right\}$.

Under the standard multi-object system model [32] described above, the Bayes recursion propagates a GLMB information state $\boldsymbol{\pi}$ to the next GLMB information state

$$\boldsymbol{\pi}_{a,+} = \Omega\left(\boldsymbol{\pi}; \boldsymbol{Z}_+, a\right), \quad (6)$$

where Ω denotes the *MS-GLMB recursion operator* (which depends on the multi-object model parameters r_B, p_B, r_S, f_+, $\kappa^{(n)}$, $P_D^{(n)}$, and $g^{(n)}$, $n \in \mathcal{N}$, see Appendix A for the actual mathematical expression). Thus, starting with an initial GLMB, all subsequent information states are GLMBs. In practice, the number of GLMB components grows with time and truncation is needed to curb this growth [24].

III. Path Planning for Multi-Object Tracking

This section presents our approach to multi-agent planning, with limited FoV sensors, for searching and tracking an unknown and time-varying number of objects, as conceptualised in Fig. 2. Subsection III-A extends the notion of differential entropy and mutual information to RFS. Subsection III-B discusses labelled multi-Bernoulli as the labelled first moment of labelled RFS and derives its differential entropy. Building on this, Subsections III-C and III-D formulate the tracking and discovery value functions. The proposed multi-agent path planning algorithm is presented in Subsections III-E and III-F.

A. Differential Entropy

Differential entropy quantifies the uncertainty of a random variable, with lower values indicating lower uncertainty [25, pp.6]. Further, in probabilistic planning, differential entropy can be used to assess the information gained from new observations [38]. Consequently, an extension of this concept to RFS is needed to formulate the information-based planning objectives for our MPOMDP, and to derive tractable solutions.

To define meaningful differential entropy for RFS, we need to revisit the notion of probability density. The probability density of an RFS is taken with respect to (w.r.t.) the reference measure μ defined for each (measurable) $\mathcal{T} \subseteq \mathcal{X}$ (the class of finite subsets of $\mathbb{X}$) by

$$\mu(\mathcal{T}) \triangleq \sum_{i=0}^{\infty} \frac{1}{i!K^i} \int 1_{\mathcal{T}}(\{y_1, \ldots, y_i\}) d(y_1, \ldots, y_i),$$

where K is the unit of hyper-volume on $\mathbb{X}$, $1_{\mathcal{T}}(\cdot)$ is the indicator function for $\mathcal{T}$, and by convention the integral for $i = 0$ is $1_{\mathcal{T}}(\emptyset)$ [39]. The role of μ is analogous to the Lebesgue measure on Euclidean space, and the integral of a function $f : \mathcal{X} \to (-\infty, \infty)$ w.r.t. μ is given by

$$\int f(Y)\mu(dY) = \sum_{i=0}^{\infty} \frac{1}{i!K^i} \int f(\{y_1, \ldots, y_i\}) d(y_1, \ldots, y_i).$$

Note that f and μ are both dimensionless/unitless. The probability density $f_X : \mathcal{X} \to [0, \infty)$ of an RFS X satisfies $\Pr(X \in \mathcal{T}) = \int 1_{\mathcal{T}}(Y) f_X(Y) \mu(dY)$ for each $\mathcal{T} \subseteq \mathcal{X}$.

The integral above is equivalent to Mahler's set integral (see Proposition 1 in [39]), defined for a function π by [32]

$$\int \pi(Y)\delta Y \triangleq \sum_{i=0}^{\infty} \frac{1}{i!} \int \pi(\{y_1, \ldots, y_i\}) d(y_1, \ldots, y_i),$$

in the sense that $\int K^{-|Y|} f(Y) \delta Y = \int f(Y)\mu(dY)$. This equivalence means that Mahler's multi-object density of the RFS X is given by $\pi_X(Y) \triangleq K^{-|Y|} f_X(Y)$.

Equipped with the above construct of density/integration, the notion of differential entropy (and mutual information) naturally extends to RFS as follows.

Definition 1: The *differential entropy* $h(X)$ of an RFS X, with probability density f_X, is defined as

$$h(X) = -\mathbb{E}_X\left[\ln f_X\right] = -\int \ln\left(f_X(Y)\right) f_X(Y) \mu(dY). \quad (7)$$

Remark 3: Differential entropy can be extended to a sequence of RFS $X_1, ..., X_m$, with joint probability density $f_{X_1,...,X_m}$ as $h(X_1, ..., X_m) = -\mathbb{E}_{X_1,...,X_m}\left[\ln f_{X_1,...,X_m}\right]$, and to conditional differential entropy of X on Z, with conditional probability density $f_{X|Z}$ as

$$h(X|Z) = -\mathbb{E}_{X,Z}\left[\ln f_{X|Z}\right]. \quad (8)$$

[3]Maps in which no two distinct arguments are mapped to the same positive value so that distinct objects cannot share the same measurement.

This notion of differential entropy means that the mutual information between the RFSs X and Z is

$$I(X;Z) = h(X) - h(X|Z). \tag{9}$$

Remark 4: Using Mahler's set integral, differential entropy can be written in terms of the multi-object density π_X as

$$h(X) = -\int \ln\left(K^{|Y|}\pi_X(Y)\right)\pi_X(Y)\delta Y. \tag{10}$$

A low differential entropy $h(X)$ translates to low uncertainty in the RFS X [25, pp.6]. Moreover, given knowledge of another RFS Z, high mutual information $I(X;Z)$ means that observing Z would provide more information (or reduce uncertainty) on X, because $I(X;Z)$ quantifies the "amount of information" obtained about X by observing Z [25, pp.6]. Hence, from a state estimation context, it is prudent to minimise the differential entropy of X, or maximise its mutual information with the observation Z.

B. Differential Entropy for Labelled Multi-Bernoulli

While differential entropy for labelled RFSs is computationally intractable in general, for the special case of *labelled multi-Bernoulli* (LMB), this can be computed analytically. An LMB, with parameters $\{r^{(\ell)}, p^{(\ell)}(\cdot)\}_{\ell\in\mathbb{L}}$, has multi-object density of the form

$$\widehat{\boldsymbol{\pi}}(\boldsymbol{X}) = r^{\mathcal{L}(\boldsymbol{X})}\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})}p^{\boldsymbol{X}}, \tag{11}$$

where $r^{(\ell)}$ is the existence probability of object ℓ, $\tilde{r}^{(\ell)} = 1 - r^{(\ell)}$, and $p(x,\ell) = p^{(\ell)}(x)$ is the probability density of its attribute conditional on existence. Like the Poisson, an LMB is *completely characterised* by the *first-moment density*, commonly known as the Probability Hypothesis Density (PHD), given by $\boldsymbol{v}(x,\ell) = r^{(\ell)}p^{(\ell)}(x)$, specifically, $\widehat{\boldsymbol{\pi}}(\boldsymbol{X}) = \widehat{\boldsymbol{\pi}}(\emptyset)\left[\boldsymbol{v}/\tilde{r}\right]^{\boldsymbol{X}}$, i.e., a multi-object exponential of the PHD (assuming $r^{(\ell)} < 1$).

Analogous to the Poisson, the LMB that matches the PHD of a labelled RFS is treated as its *labelled first moment*, e.g., the labelled first moment of the GLMB $\boldsymbol{\pi} \triangleq \left\{\left(w^{(I,\xi)}, p^{(\xi)}\right)\right\}$ is the LMB with [26]

$$r^{(\ell)} = \sum_{I,\xi} 1_I(\ell)w^{(I,\xi)}, \tag{12}$$

$$p^{(\ell)}(x) = \sum_{I,\xi} \frac{1_I(\ell)w^{(I,\xi)}p^{(\xi)}(x,\ell)}{r^{(\ell)}} \tag{13}$$

The key benefits of the LMB over the Poisson (unlabeled first moment) are the trajectory information and a cardinality variance that does not grow with the mean.

The following Proposition establishes an analytic expression for the differential entropy of the LMB, which can be evaluated with $\mathcal{O}(|\mathbb{L}|)$ complexity, i.e., linear in the number of labels. The proof is given in Appendix B.

Proposition 1: The differential entropy of an LMB $\boldsymbol{X}$, with parameter set $\{r^{(\ell)}, p^{(\ell)}(\cdot)\}_{\ell\in\mathbb{L}}$ is

$$h(\boldsymbol{X}) = -\sum_{\ell\in\mathbb{L}}\Big[r^{(\ell)}\ln r^{(\ell)} + \tilde{r}^{(\ell)}\ln\tilde{r}^{(\ell)} + r^{(\ell)}\langle p^{(\ell)}, \ln(Kp^{(\ell)})\rangle\Big]. \tag{14}$$

Remark 5: A special case of Proposition 1 is the differential entropy of a Bernoulli RFS, *i.e.*, an LMB with only one component, parameterised by (r,p):

$$h(X) = -\big[r\ln(r) + \tilde{r}\ln\tilde{r} + r\langle p, \ln(Kp)\rangle\big].$$

C. Tracking Value Function

This subsection presents the information-based value function for the tracking task. Choosing the action that minimises the mutual information between the multi-object state and the observation (resulting from the action) reduces uncertainty on the multi-object state and, hence, improves tracking accuracy. Note from (9) that the mutual information between the RFSs X and Z is $I(X;Z) = h(X) - h(X|Z)$. Since $h(X)$ is independent of the control action a, maximising the mutual information $I(X;Z)$ is equivalent to minimising $h(X|Z)$, the differential entropy of X conditioned on Z.

For computational tractability, we adopt the PIMS approach [32]. Specifically, we use a tracking value function $V_1(\cdot)$ of the form (4), with immediate reward, at time j, given by $\boldsymbol{\varrho}_j\big(\widehat{\boldsymbol{Z}}_{a,j}, \boldsymbol{X}_j, a\big) = \ln\big(K^{|\boldsymbol{X}|}\widehat{\boldsymbol{\pi}}_{a,j}(\boldsymbol{X}_j|\widehat{\boldsymbol{Z}}_{a,j})\big)$, and $T\big(\boldsymbol{\pi}_{a,j}(\boldsymbol{X}_j|\widehat{\boldsymbol{Z}}_{a,j})\big) = \widehat{\boldsymbol{\pi}}_{a,j}(\boldsymbol{X}_j|\widehat{\boldsymbol{Z}}_{a,j})$, where $\widehat{\boldsymbol{\pi}}_{a,j}(\cdot|\widehat{\boldsymbol{Z}}_{a,j})$ denotes the labelled first moment of the information state $\boldsymbol{\pi}_{a,j}(\cdot|\widehat{\boldsymbol{Z}}_{a,j})$, i.e., $\widehat{\boldsymbol{\pi}}_{a,j}(\cdot|\widehat{\boldsymbol{Z}}_{a,j})$ is the LMB matching the first moment of $\boldsymbol{\pi}_{a,j}(\cdot|\widehat{\boldsymbol{Z}}_{a,j})$. This results in

$$V_1(a) = -\sum_{j=k+1}^{k+H} h(\boldsymbol{X}_j|\widehat{\boldsymbol{Z}}_{a,j}), \tag{15}$$

i.e., the cumulative differential entropy of the multi-object state given the PIMS, over the horizon.

It is important to note that, in addition to being a meaningful tracking value function, $V_1(\cdot)$ can be evaluated analytically and efficiently. Since the information state is a GLMB (see Subsection II-C), the labelled first moment is the LMB $\widehat{\boldsymbol{\pi}}_{a,j}(\boldsymbol{X}_j|\widehat{\boldsymbol{Z}}_{a,j})$ whose parameters are given by (12)-(13). Moreover, the differential entropy of this LMB can be evaluated in closed form, as given in (14), with a complexity that is *linear* in the number of labels, i.e., $\mathcal{O}(|\mathbb{L}|)$. This linear complexity is achieved because each LMB component can be updated individually as the ideal measurement has a known origin and no false positives/negatives. The complexity of this step is $\mathcal{O}(1)$. In addition, due to **Proposition 1**, the complexity of computing the differential entropy is $\mathcal{O}(|\mathbb{L}|)$. Hence, the total complexity of computing V_1 is $\mathcal{O}(|\mathbb{L}|)$. This is a significant improvement in complexity compared to employing other multi-object filters.

Remark 6: Using the expected value function (3) with

$$\begin{aligned}\boldsymbol{\varrho}_j(\boldsymbol{Z}, \boldsymbol{X}, a_{j-1}) = &\ln\big(K^{|\boldsymbol{X}|}\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}|\boldsymbol{Z})\big)\\ &- \ln\int K^{|\boldsymbol{X}|}\boldsymbol{\pi}_{a_{j-1},j}(\boldsymbol{X}|\boldsymbol{Y})\boldsymbol{g}_j(\boldsymbol{Z}|a_{j-1})\delta\boldsymbol{Y},\end{aligned}$$

yields the cumulative mutual information between the multi-object state and its observations over the horizon. However, as discussed earlier, this value function is intractable to evaluate.

D. Occupancy-based Discovery Value Function

This subsection presents the entropy-based value function for the discovery task via an occupancy grid. Intuitively, the knowledge of unexplored regions can be incorporated into the planning to increase the likelihood of discovering new objects. While objects of interest outside the sensor's FoVs are undetected, this knowledge has not been exploited for discovery. To this end, we develop a dynamic occupancy grid to capture knowledge of undetected objects outside the sensor's FoVs.

We partition the search area $G \subset \mathbb{R}^{d_{\varkappa}}$ into a grid $\{\varkappa^{(i)}\}_{i=1}^{N}$, such that $\varkappa^{(i)} \cap \varkappa^{(j)} = \emptyset, i \neq j$, and $G = \varkappa^{(1)} \cup \cdots \cup \varkappa^{(N)}$. At time k, the *occupancy* of grid cell $\varkappa^{(i)}$ is modelled as a Bernoulli random variable $O^{(i)}$, i.e., $O^{(i)} = 1$ means cell $\varkappa^{(i)}$ is occupied, and $O^{(i)} = 0$ otherwise. Further, let $\boldsymbol{Z}_{a_-}(\varkappa^{(i)})$ denotes the set of multi-agent measurements originating from cell $\varkappa^{(i)}$ after action a_- was taken at the previous time, and $Y_{a_-}^{(i)} = \delta_\emptyset[\boldsymbol{Z}_{a_-}(\varkappa^{(i)})]$ denotes the binary observation, which equals 1 if no measurements originate from cell $\varkappa^{(i)}$, and 0 otherwise. We define the discovery value function V_2, for the PIMS approach, as the cumulative differential entropy (which is also the Shannon entropy when the random variable is discrete) of the occupancy grid over the horizon

$$V_2(a) = -\sum_{j=k+1}^{k+H} \sum_{i=1}^{N} h(O_j^{(i)} | Y_{a,j}^{(i)}). \tag{16}$$

Intuitively, for discovery, we are only interested in the occupancy of cells in which objects are undetected, because cells with detected objects have already been explored. The following result shows that the entropy $h(O_j^{(i)}|Y_{a,j}^{(i)})$ only depends on $\omega_j^{(i)}(a) \triangleq \Pr(O_j^{(i)} = 1 | Y_{a,j}^{(i)} = 1)$, i.e., the probability that $\varkappa^{(i)}$ is occupied by undetected objects. Moreover, the discovery value function (16) can be computed analytically by propagating an initial $\omega_k^{(i)}(a_{k-1})$ from k to $j = k+1, ..., k+H$, for each cell. The proof is given in Appendix C.

Proposition 2: Let $P_{S,+}^{(i)}$ be the probability that at least one undetected object in $\varkappa^{(i)}$ is still there at the next time, $P_{B,+}^{(i)}$ be the probability of at least one undetected object entering the cell at the next time, and $Q_{D,+}^{(i)}(a)$ be the probability that objects in cell $\varkappa^{(i)}$ will not be detected by any agent at the next time if action a is taken at the current time. Then,

$$\begin{aligned} h(O_j^{(i)}|Y_{a,j}^{(i)}) = &-\Big[1 - \omega_{j-1}^{(i)}(a) + \omega_{j-1}^{(i)}(a) Q_{D,j}^{(i)}(a)\Big] \\ &\times \Big[(1-\omega_j^{(i)}(a)) \ln(1-\omega_j^{(i)}(a)) + \omega_j^{(i)}(a) \ln \omega_j^{(i)}(a)\Big]. \end{aligned} \tag{17}$$

where $Q_{D,j}^{(i)}(a) = \prod_{n \in \mathcal{N}} (1 - P_{D,j}^{(i)}(a^{(n)}))$, and $P_{D,j}^{(i)}(a^{(n)})$ is the probability that agent n detects objects in cell $\varkappa^{(i)}$ at time j when control action $a^{(n)}$ is taken. In other words, the probability of objects in cell $\varkappa^{(i)}$ are not detected by any agents depends on detection probabilities of cell $\varkappa^{(i)}$ from all agents for a given control action. Moreover, given $\omega_j^{(i)}(a_{j-1})$, the next $\omega_{j+1}^{(i)}(a_j)$ is given by

$$\omega_{j+1}^{(i)}(a_{j-1}) = ((1 - \omega_j^{(i)}(a_{j-1})) P_{B,+}^{(i)} + \omega_j^{(i)}(a_{j-1}) P_{S,+}^{(i)}, \tag{18}$$

$$\omega_{j+1}^{(i)}(a_j) = \frac{\omega_{j+1}^{(i)}(a_{j-1}) Q_{D,+}^{(i)}(a_j)}{1 - \omega_{j+1}^{(i)}(a_{j-1}) + \omega_{j+1}^{(i)}(a_{j-1}) Q_{D,+}^{(i)}(a_j)}. \tag{19}$$

Selecting the action(s) that minimises the differential entropy of the occupancy grid (i.e., maximise V_2) improves the likelihood of discovering previously undetected objects. Recently explored cells would have low entropy due to observations made by the agents. In contrast, the entropy of unexplored cells would be higher because of the lack of observations. Hence, actions that minimise the differential entropy of the occupancy grid tend to drive the agents to the unexplored cells.

E. Multi-Objective Planning

Multi-agent path planning with the competing objectives of discovery and tracking can be naturally fulfilled by multi-objective optimisation via the multi-objective value function

$$V(a) = [V_1(a), V_2(a)]^\dagger, \tag{20}$$

where $a \in \mathbb{A}$, V_1 and V_2 are respectively the tracking and discovery value functions described in (15) and (16). Multi-objective optimisation identifies meaningful trade-offs amongst objectives via the Pareto-set, wherein no solutions can improve one objective without degrading the remaining ones.

Since online planning requires finding a solution in a timely manner, we adopt the global criterion method (GCM) [40]. This is one of the standard computational techniques in multi-objective optimisation, which computes a trade-off solution based on the distance of the value functions from an ideal solution. In particular, GCM combines the two value functions into one by first normalising them, and then combining the normalised versions without weighting. Hence, GCM avoids pre-defining specific weights for each value function, which simplifies the decision-making process, leading to faster solutions. In scenarios without human intervention or preference information, GCM's equal weighting provides a neutral trade-off solution, avoiding bias towards any specific value function. Importantly, if the underlying value functions are submodular, GCM *preserves* the submodularity property. The resulting value function for GCM, given by

$$V_{mo}(a) = \sum_{i=1}^{2} \frac{V_i(a) - \min_{a \in \mathbb{A}} V_i(a)}{\max_{a \in \mathbb{A}} V_i(a) - \min_{a \in \mathbb{A}} V_i(a)}, \tag{21}$$

yields a unique solution [41], and turns the multi-objective optimisation problem into

$$\max_{a \in \mathbb{A}} V_{mo}(a). \tag{22}$$

In principle, solving problem (22) is NP-hard [18]. Nonetheless, when V_{mo} is *monotone submodular* (this holds when V_1 and V_2 are monotone submodular), sub-optimal solutions with guaranteed optimality bound can be computed using a greedy algorithm with drastically lower complexity [42], [43]. An

Algorithm 1 Greedy Search

```
1: Input: 𝓐, V_mo                              ▷ action space and value function.
2: Output: A^g ∈ 𝓐                              ▷ greedy control action.
3: A^g := ∅                                     ▷ initialise empty greedy control action.
4: P := ∅                                       ▷ initialise empty planned agent's list.
5: U := 𝒩                                       ▷ initialise planning agent's list.
6: while U ≠ ∅ do
7:     for each n ∈ U do
8:         [a_g^(n), V^(n)] := argmax_{a^(n) ∈ 𝔸^(n)} V_mo(A^g ⊎ a^(n))
9:     end for
10:    n* := argmax_{n∈U} V^(n)                 ▷ n* that yields the best value function.
11:    A^g := A^g ⊎ a_g^(n*)                    ▷ store action with agent n*.
12:    P := P ∪ {n*}                            ▷ add n* into planned agent's list.
13:    U := U \ {n*}                            ▷ delete n* from planning agents' list.
14: end while
```

alternative to GCM is to seek a robust solution against the worst possible objective [44]. However, this robust submodular observation selection (RSOS) usually results in a non-submodular value function. Another alternative is a weighted sum of the value functions, which preserves submodularity, but choosing a meaningful set of weights is an open problem.

F. Greedy Search Algorithm

Greedy search can provide suboptimal solutions to the MPOMDP in polynomial-time with guaranteed optimality bound if the objective *is monotone submodular* [42], [43].

Definition 2: A set function q is said to be *monotone* if $\boldsymbol{A} \subseteq \boldsymbol{B} \Rightarrow q(\boldsymbol{A}) \leq q(\boldsymbol{B})$, and *submodular* if $\boldsymbol{A} \subseteq \boldsymbol{B}$ and $\boldsymbol{a} \notin \boldsymbol{B} \Rightarrow q(\boldsymbol{B} \cup \{\boldsymbol{a}\}) - q(\boldsymbol{B}) \leq q(\boldsymbol{A} \cup \{\boldsymbol{a}\}) - q(\boldsymbol{A})$ [45].
To show monotone submodularity of the proposed value functions, we recast them as set functions on the partition matroid $\boldsymbol{\mathcal{A}}$ of the *common action space* $\mathbb{A} \triangleq \uplus_{n\in\mathcal{N}} \mathbb{A}^{(n)} \times \{n\}$. For each $n \in \mathcal{N}$, recall that $V_j((a^{(1)}, \ldots, a^{(n)}))$, $j = 1, 2, mo$, are the relevant value functions on the multi-agent action space $\mathbb{A}^{(1)} \times \ldots \times \mathbb{A}^{(n)}$, we define corresponding value functions $\boldsymbol{V}_j(\cdot)$ on $\boldsymbol{\mathcal{A}}$ by

$$\boldsymbol{V}_j(\{(a^{(m)}, m)\}_{m=1}^{n}) \triangleq V_j((a^{(1)}, \ldots, a^{(n)})).$$

Note that the subset $\boldsymbol{\mathcal{A}}^{(n)} = \{\boldsymbol{A} \in \boldsymbol{\mathcal{A}} : |\boldsymbol{A}| = n\}$ of $\boldsymbol{\mathcal{A}}$ is equivalent to $\mathbb{A}^{(1)} \times \ldots \times \mathbb{A}^{(n)}$ because any multi-agent action $(a^{(1)}, \ldots, a^{(n)})$ has a 1-1 correspondence with $\{(a^{(m)}, m)\}_{m=1}^{n} \in \boldsymbol{\mathcal{A}}^{(n)}$. Thus, problem (22) is equivalent to maximising $\boldsymbol{V}_{mo}(\boldsymbol{A})$ over the partition matroid $\boldsymbol{\mathcal{A}}$ subject to the cardinality constraint $|\boldsymbol{A}| = |\mathcal{N}|$. Here, using a set function facilitates the analysis of the greedy search algorithm (Algorithm 1) that iteratively appends one single agent at each step (line 11).

Proposition 3: The mutual information $I\big(\boldsymbol{X}; \boldsymbol{Z}_{\boldsymbol{A}}\big)$ between the multi-object state $\boldsymbol{X}$ and multi-agent measurement $\boldsymbol{Z}_{\boldsymbol{A}}$ collected by the agents under action $\boldsymbol{A} \in \boldsymbol{\mathcal{A}}$, is a monotone submodular set function on $\boldsymbol{\mathcal{A}}$ (*see Appendix D for proof*).

The above Proposition implies that the conditional differential entropy $-h(\boldsymbol{X}|\boldsymbol{Z}_{\boldsymbol{A}})$ is also a monotone submodular set function, because $I\big(\boldsymbol{X}; \boldsymbol{Z}_{\boldsymbol{A}}\big) = h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{Z}_{\boldsymbol{A}})$ and $h(\boldsymbol{X})$ is independent of $\boldsymbol{A}$. Moreover, since $\boldsymbol{V}_1(\boldsymbol{A})$ is a positive linear combination of $-h(\boldsymbol{X}|\widehat{\boldsymbol{Z}}_{\boldsymbol{A}})$, using [45, pp.272], we have the following result.

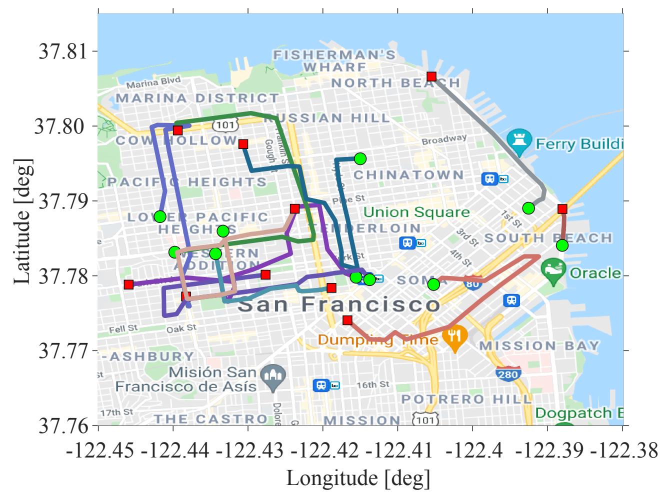

Fig. 3. CRAWDAD taxi dataset: 10 taxis travelling over a 1000 s period, within an area of 6150 m×6080 m. The symbols ◯/☐ indicate the Start/Stop positions of each taxi, the different colours representing different taxis

Proposition 4: The tracking value function $\boldsymbol{V}_1$ is a monotone submodular set function.

In addition, the same result also holds for the discovery value function (*see Appendix D for proof*).

Proposition 5: The discovery value function $\boldsymbol{V}_2$ is a monotone submodular set function.

It follows from the above propositions that the objective $\boldsymbol{V}_{mo}$ is also a monotone submodular function on $\boldsymbol{\mathcal{A}}$, because it is a positive linear combination of $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ [45, pp.272]. This means that the inexpensive Greedy Search (see **Algorithm 1**) can be used to compute a suboptimal solution, with the following optimality bound [46].

Proposition 6: Let $\mathring{\boldsymbol{V}}_{mo} = \max_{\boldsymbol{A}\in\boldsymbol{\mathcal{A}}, |\boldsymbol{A}|=|\mathcal{N}|} \boldsymbol{V}_{mo}(\boldsymbol{A})$, and $\boldsymbol{A}^*$ denote a solution computed via the Greedy Search Algorithm 1. Then

$$(1 - e^{-1})\mathring{\boldsymbol{V}}_{mo} \leq \boldsymbol{V}_{mo}(\boldsymbol{A}^*) \leq \mathring{\boldsymbol{V}}_{mo}. \tag{23}$$

Remark 7: Computing the optimal multi-agent control action via exhaustive search incurs an $\mathcal{O}(|\mathbb{A}|^{|\mathcal{N}|}|\mathcal{N}|(|\mathbb{L}| + N)H)$ complexity. The above Proposition enables suboptimal solutions to be computed via greedy search with a tight optimality bound, but at a reduced $\mathcal{O}\left(|\mathbb{A}||\mathcal{N}|^2(|\mathbb{L}| + N)H\right)$ complexity. This is a drastic reduction from exponential complexity to linear in the number of control actions, and quadratic in the number of agents.

IV. Performance Evaluations

This section demonstrates the performance of our proposed multi-agent solution via numerical experiments. Subsections IV-A and IV-B present the results and discussions of simulated experiments on the CRAWDAD dataset of taxi trajectories in the San Francisco Bay Area [27]. Further tests on simple yet challenging hypothetical scenarios are presented in Subsection IV-C. Simulations were run on a workstation with two AMD EPYC 7702 processors (each with 64 cores @ 2.0 GHz), 1 TB of RAM, and MATLAB 2022a.

A. CRAWDAD Dataset Experiment Settings

The CRAWDAD dataset's real-world taxi trajectories, coupled with synthetic measurements, allow us to control various experimental parameters, especially with the time-varying number of agents and objects. In particular, we randomly selected 10 taxi tracks over 1000 s (from 18-May-2008 4:43:20 PM to 18-May-2008 5:00:00 PM), as shown in Fig. 3. Following [19], the 6150 m $\times$ 6080 m search area is scaled down by a factor of 5, while the time is sped up by 5 so that the taxi's speed remains the same as in the real world. Thus, the simulated environment has an area of 1230 m $\times$ 1216 m and a total search time of 200 s.

Since taxis can turn into different streets, we employ a constant turn (CT) model with an unknown turn rate to account for this. In particular, let $\boldsymbol{x} = (x, \ell)$ denote the single object state comprising the label ℓ and kinematics $x = [\tilde{x}^\dagger, \theta]^\dagger$ where $\tilde{x} = [\rho_x, \dot{\rho}_x, \rho_y, \dot{\rho}_y]^\dagger$ is its position and velocity in Cartesian coordinates, and θ is the turn rate. Each object follows the constant turn model given by

$$\tilde{x}_{k+1|k} = F^{CT}(\theta_k)\tilde{x}_k + G^{CT}\eta_k, \quad (24)$$

$$\theta_{k+1|k} = \theta_k + T_0 q_k \quad (25)$$

where

$$F^{CT}(\theta) = \begin{bmatrix} 1 & \dfrac{\sin(\theta T_0)}{\theta} & 0 & -\dfrac{1-\cos(\theta T_0)}{\theta} \\ 0 & \cos(\theta T_0) & 0 & -\sin(\theta T_0) \\ 0 & \dfrac{1-\cos(\theta T_0)}{\theta} & 1 & \dfrac{\sin(\theta T_0)}{\theta} \\ 0 & \sin(\theta T_0) & 0 & \cos(\theta T_0), \end{bmatrix}, \quad (26)$$

$$G^{CT} = \begin{bmatrix} T_0^2/2 & T_0 & 0 & 0 \\ 0 & 0 & T_0^2/2 & T_0 \end{bmatrix}^\dagger, \quad (27)$$

$T_0 = 1$ s is the sampling interval, $\eta_k \sim \mathcal{G}(\cdot; 0, 0.15^2 I_2)$ is Gaussian noise with I_2 denoting the 2×2 identity matrix, and $q_k \sim \mathcal{G}(\cdot; 0, (\pi/60)^2)$.

The sensor on each agent n is range-limited to r_D and its detection probability follows:

$$P_D^{(n)}(\boldsymbol{x}, u^{(n)}) = \begin{cases} P_D^{\max} & d(\boldsymbol{x}, u^{(n)}) \le r_D, \\ \max(0, P_D^{\max} - (d(\boldsymbol{x}, u^{(n)}) - r_D)\hbar) & \text{otherwise}, \end{cases} \quad (28)$$

where $d(\boldsymbol{x}, u^{(n)})$ is the distance between the object $\boldsymbol{x}$ and agent $u^{(n)}$. A *position* sensor is considered in our experiments wherein a detected object $\boldsymbol{x}$ yields a noisy position measurement $z = [\rho_x, \rho_y]^\dagger + v$, with $v \sim \mathcal{G}(\cdot; 0, R)$ and $R = \mathrm{diag}([\sigma_x^2, \sigma_y^2])$. The parameters for our experiments are selected according to real-world settings in [47]. The detection range $r_D = 126$ m and $\hbar = 0.008$, while the false-alarm per scan λ_c is 0.0223, and the maximum detection probability $P_D^{\max} = 0.8825$ (see Table VIII in [48]). The minimum altitude of the UAVs is 126 m, and to ensure no collisions, each operates at a different height with a 5 m vertical separation. We observe that the estimation error from a UAV flying at 60 m altitude using a standard visual sensor is around 0.55 m [47]. Since our UAVs fly at higher, and the measurement noise is proportional to the UAV's altitude, we set the measurement noise $\sigma_x = \sigma_y = 0.55 \times 126/60 = 1.115$ m.

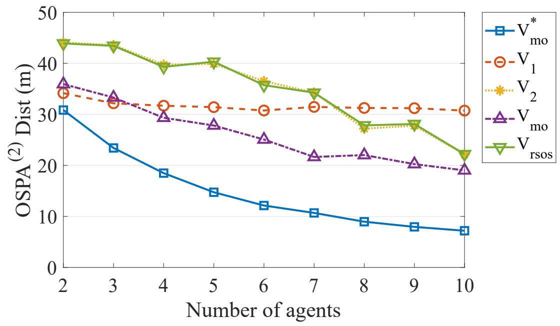

Fig. 4. Tracking performance of multi-objective planning with V_{mo} versus baseline methods in 100 Monte Carlo runs as $|\mathcal{N}|$ increases from 2 to 10. V_{mo}^* represents the optimal performance under ideal conditions where measurement origins are known.

A maximum number of 10 UAVs (*e.g.*, quad-copters) is considered in this experiment. These UAVs depart from the centre of the search area, *i.e.*, $[615, 608, 126]^\dagger$. The control action space $\mathbb{A}^{(n)}$ is $\{\leftarrow, \nwarrow, \uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \circ\}$, $\forall n \in \mathcal{N}$, which represents the prescribed headings of the drone while moving with a maximum speed of 20 m/s or staying stationary ($\circ$) at the current position. Since there is no prior knowledge of each object's state, we model initial births at time $k = 0$ by an LMB density with parameters $\{(r_{B,0}^{(i)}, p_B^{(i)})\}_{i=1}^{N_B}$, where $N_B = 100$ is the number of possible new births, $p_B^{(i)} = \mathcal{G}(\cdot; m_B^{(i)}, \Sigma_B)$, with $m_B^{(i)} = [m_{B,x}^{(i)}, 0, m_{B,y}^{(i)}, 0]^\dagger$, and $\Sigma_B = \mathrm{diag}([\Delta_x/2, 1, \Delta_y/2, 1])^2$. In this experiment we use $m_{B,x}^{(i)} = \Delta_x/2 + ((i-1) \mathrm{mod}\ 10)\Delta_x$, $m_{B,y}^{(i)} = \Delta_y/2 + \lfloor i/10 \rfloor \Delta_y$, $\Delta_x = 123$ m, and $\Delta_y = 121.6$ m, with "mod" denoting the modulo operator and $\lfloor\cdot\rfloor$ denoting the floor operator. For the next time step, we use an adaptive birth procedure that incorporates the current grid occupancy information at time k into the birth probability at the next time $k+1$. Note that, since the number of occupancy grid cells N can be significantly large ($10,000$ in this case), which increases the filtering time if there are too many birth components, we propose resizing the grid resolution from N cells with occupancy probability $\{\omega^{(i)}(a)\}_{i=1}^{N}$ to N_B cells with occupancy probability $\{\bar{\omega}^{(i)}\}_{i=1}^{N_B}$ where $N_B \ll N$, using bicubic interpolation (e.g., with the imresize command in MATLAB) to efficiently improve the filtering time. The birth existence probability is then updated by:

$$r_{B,+}^{(i)} = \frac{\left[1 + \bar{\omega}^{(i)}/\max(\bar{\omega}^{(i)})\right]\left(\sum_{i=1}^{N_B} r_{B,0}^{(i)}\right)}{\sum_{i=1}^{N_B}\left[1 + \bar{\omega}^{(i)}/\max(\bar{\omega}^{(i)})\right]}, \quad (29)$$

to ensure that the number of births remains stable over time, i.e., maintaining a constant expected number of births $\sum_{i=1}^{N_B} r_{B,+}^{(i)}$.

To evaluate tracking performance, we use the optimal subpattern assignment (OSPA$^{(2)}$) metric from [49] with a cut-off value of $c = 50$ m and order $p = 1$ over a window spanning the entire experimental duration. All results are averaged over 100 Monte Carlo runs. A smaller **OSPA$^{(2)}$ Dist (m)** indicates a better tracking performance, covering localisation accuracy, cardinality, track fragmentation, and track switching errors.

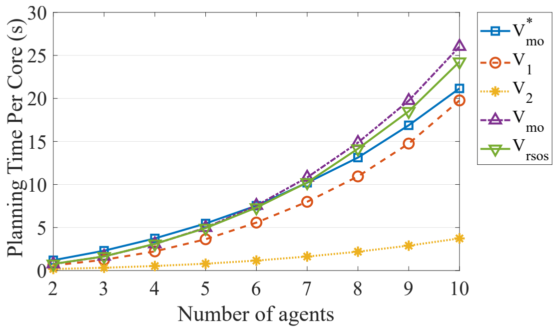

Fig. 5. Average planning time of multi-objective planning with V_{mo} versus baseline methods in 100 Monte Carlo runs as $|\mathcal{N}|$ increases from 2 to 10.

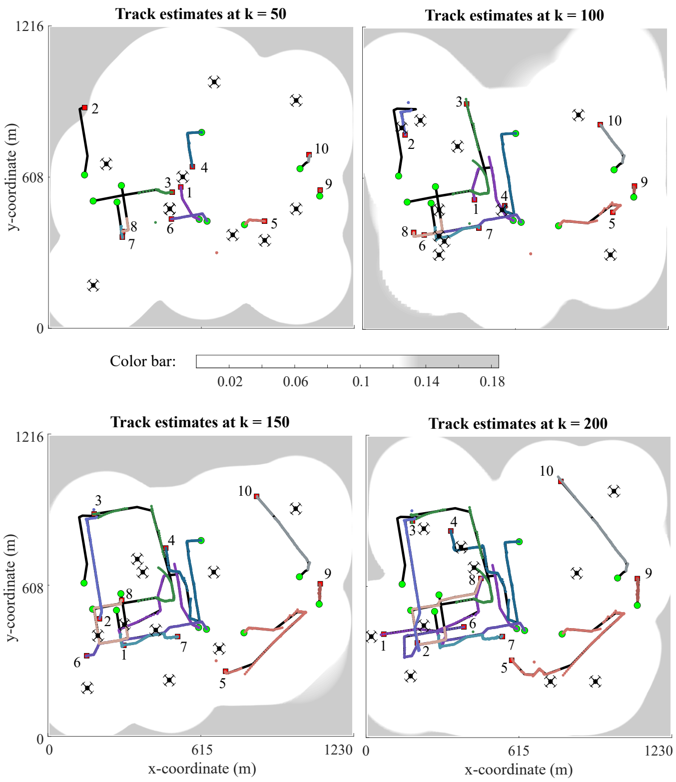

Fig. 6. A sample run for multi-objective planning with V_{mo}. **Background**: grid occupancy probability. **Foreground**: estimated and true positions of 10 taxis. Black lines show the ground truth trajectories, while the coloured dots denote the estimated positions. Circles (◯) and squares (☐) mark taxi start/stop points, with different colours representing different taxi identities. The drone symbol indicates the positions of the 10 UAVs.

B. Results and Discussions

We first examine how the performance varies when the number of agents increases, specifically, for the following three planning strategies: ***(i)*** *tracking* only objective function V_1 (conventional approach); ***(ii)*** *discovering* only objective function V_2 (a special case of our approach); and ***(iii)*** multi-objective value function V_{mo} that optimises trade-offs between tracking and discovering tasks (our approach). Note that, the value function V_1 is a conventional information-based method used in several previous works [19], [50]–[54], and hence, considered as a baseline for comparisons. Additionally, we benchmark the planning algorithms against the ideal case when data association is known [28], i.e., the best possible tracking performance. Furthermore, we compare V_{mo} versus $V_{rsos} = \min(V_1, V_2)$, i.e., the robust submodular observation

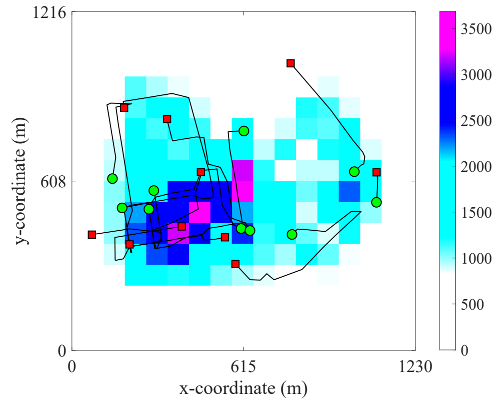

Fig. 7. Heatmap (16 × 16 grids) showing the positions of 10 UAVs and taxi trajectories in multi-objective planning with V_{mo}, over 100 MC runs. Circles (◯) and squares (☐) indicate taxi start/stop positions. The colour legend represents UAV position frequency.

TABLE II
PERFORMANCE COMPARISON FOR DIFFERENT DETECTION PROBABILITIES AND CLUTTER RATES WHEN $|\mathcal{N}| = 10$

		OSPA$^{(2)}$Dist [m]			
$P_D^{\max}$	λ_c	V_1	V_2	V_{mo}	V_{mo}^*
0.9	10	26.7	23.6	**18.1**	10.1
	20	28.7	25.6	**19.6**	11.3
	30	29.5	27.3	**21.9**	12.0
0.8	10	29.3	29.9	**20.0**	12.4
	20	31.8	31.3	**22.1**	13.8
	30	32.3	31.8	**24.5**	15.1
0.7	10	31.7	31.3	**23.4**	15.4
	20	33.8	31.4	**24.7**	17.4
	30	35.6	33.2	**27.5**	18.8

selection (RSOS) method [44] (ironically the RSOS method yields a *non-submodular* value function).

Fig. 4 shows the **OSPA$^{(2)}$** tracking error, over 100 Monte Carlo (MC) runs, for 10 taxis in the CRAWDAD taxi dataset when the number of agents is increased from 2 to 10. On the one hand, when the number of agents is large (more than four), our proposed (multi-objective planning with) V_{mo} constantly outperforms (single-objective planning with) V_1 and V_2, since there are enough agents to perform both tracking and discovering tasks simultaneously. On the other hand, when the number of agents is small (less than four), V_{mo} achieves similar accuracy as V_1 since there are not enough agents to cover the area, the multi-agent focused on tracking instead of exploring. As expected, V_1 does not improve tracking performance when the number of agents increases since it only focuses on tracking detected objects and misses objects outside of the team's FoVs. Overall, multi-objective planning with V_{mo} outperforms single-objective planning that either focuses on tracking, i.e., V_1, or discovering, i.e., V_2. We also observe that V_{mo} significantly outperforms V_{rsos}. Moreover, since V_{rsos} is not sub-modular, there is no performance bound for optimising V_{rsos} using the greedy algorithm. As a result, we will *exclude* V_{rsos} from other experimental results.

Fig. 5 presents the averaged planning time using a single-core over 100 MC runs for different value functions as the number of agents increases from 2 to 10. As expected, V_2 exhibits the fastest planning time because it solely focuses on discovering new objects via the occupancy grid. In contrast, V_{mo} incurs the longest planning time as it requires the computation of both V_1 and V_2. The result further supports the efficacy of our proposed method, with planning time increasing proportionately to the square of the number of agents.

Table II presents the tracking performance for various detection probabilities and clutter rates when the number of agents is fixed at 10, across 100 Monte Carlo (MC) simulations. The results substantiate the robustness of our proposed method (V_{mo}), which performs consistently better across various detection probabilities and clutter rates. Further results for the 10-agent case demonstrate the effectiveness of multi-objective planning. Fig. 6 depicts the estimated versus true trajectories of the 10 taxis (from the CRAWDAD taxi dataset), and the occupancy probability, at various times, for a particular run of the proposed (multi-objective planning with) V_{mo}. The results demonstrate the correct search and tracking of all taxis. The heat map over 100 MC runs in Fig. 7, indicates that the 10 agents concentrate on the western region of the search area where there are more taxis around, while also slightly covering the eastern region to successfully track the remaining taxis. It is expected that as time increases, the agents start to spread out from the centre to cover more areas for better discovery and tracking.

The mean and standard deviation (over 100 MC runs) of the OSPA$^{(2)}$ and cardinality errors for the different planning methods in Fig. 8, shows that the proposed V_{mo} consistently outperforms V_1 (tracking-only) or V_2 (discovering-only) in terms of the overall tracking performance (*i.e.*, OSPA$^{(2)}$ Dist). Further, for localisation (*i.e.*, OSPA$^{(2)}$ Loc), the performance of V_{mo} approaches that of the ideal case V^*_{mo}. We observe that most of the planning methods (in the experiment) can correctly estimate the number of taxis (*i.e.*, Cardinality), except V_1, which only focuses on tracking and does not explore the areas outside the team's FoVs. In terms of OSPA$^{(2)}$ Card, since the cardinality is estimated correctly, most OSPA$^{(2)}$ Card errors come from the track label switching and track fragmentation, which is expected given the limited FoVs, and the motion model mismatch between the CT model and the taxi's occasional turns.

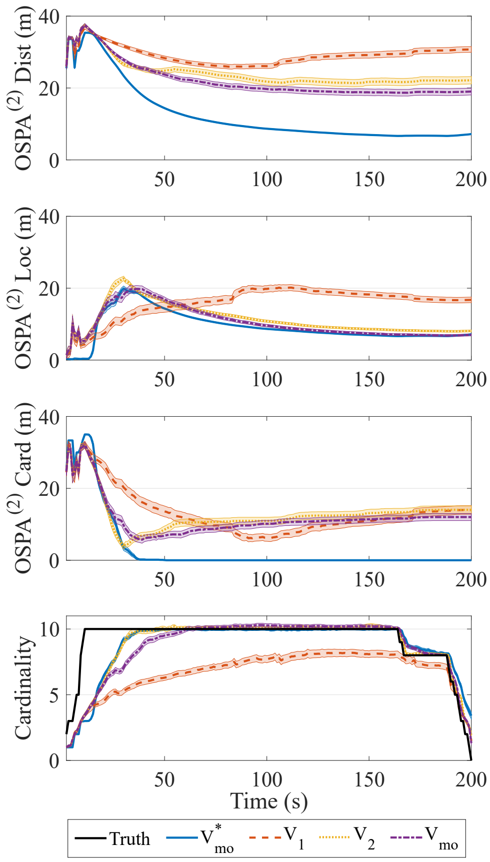

Fig. 8. Overall tracking performance of the 10 agents versus time over 100 MC runs (mean ±0.2 standard deviation), for different planning methods.

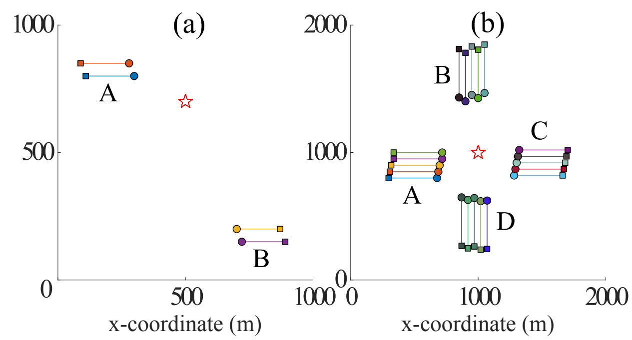

Fig. 9. Experimental settings for two simulated scenarios. The red ⋆ symbol denotes the agents' initial positions, while the symbols ◯/□ denote the Start/Stop positions of each object. (a) Scenario 1: two groups, each containing 2 objects, in a 1000 m ×1000 m region, moving outwards in opposite directions. Due to the sensor's limited detection range, locating group B, positioned beyond the detection range, solely relies on exploration. (b) Scenario 2: Four groups, each containing 5 objects, in a 2000 m ×2000 m region, moving outwards rapidly in opposite directions.

C. Challenging Hypothetical Scenarios

To highlight the differences between the value functions, we test them on two challenging hypothetical scenarios from [28], illustrated in Fig. 9.

Similar to the previous experiment, the objects move independently in a 2D plane, but according to a constant velocity model $x_{k+1|k} = F^{CV} x_k + q_k^{CV}$, where $F^{CV} = [1, T_0; 0, T_0] \otimes I_2$, $T_0 = 1$ s is the sampling interval, $\otimes$ is the Kronecker product, I_2 is the 2×2 identity matrix, $q_k^{CV} \sim \mathcal{G}(\cdot; 0, \Sigma^{CV})$ is a 4D vector of zero-mean Gaussian process noise, with covariance $\Sigma^{CV} = \sigma^2_{CV}[T_0^3/3, T_0^2/2; T_0^2/2, T_0] \otimes I_2$. We also use the same *adaptive birth* model as the previous experiment, with a constant initial birth probability $r^{(i)}_{B,0} = 0.01$, $i = 1:100$, and $\Delta_x = \Delta_y = 100$ m for Scenario 1, and $\Delta_x = \Delta_y = 200$ m for Scenario 2. A 100×100 occupancy grid (totalling $10,000$) cells) is used for both scenarios. Scenario 1 has a finer grid resolution with 10 m×10 m cells, while Scenario 2 has a coarser grid with 20 m×20 m cells.

Similar to the previous experiment, to avoid collisions, the agents operate at varying heights, with a minimum elevation of 30 m and a 5 m vertical separation between drones. Each agent has an on-board position sensor with a limited detection range of $r_D = 200$ m. The probability of detecting an object depends on its distance to the agent, and follows (28) with $P_D^{\max} = 0.9$, $\hbar = 0.008$ m^{-1}. When an object $\boldsymbol{x}$ is detected, the sensor registers a noisy position measurement $z = [\rho_x, \rho_y]^\dagger + v$,

TABLE III
AVERAGED TRACKING PERFORMANCE OVER 100 MC RUNS

Agents	Value Func.	Scenario 1 OSPA$^{(2)}$ Dist (m)	Scenario 1 Entropy (nats)	Scenario 2 OSPA$^{(2)}$ Dist (m)	Scenario 2 Entropy (nats)
$\lvert\mathcal{N}\rvert = 3$	V_1	29.0	0.22	34.1	0.27
	V_2	29.3	**0.06**	35.5	**0.20**
	V_{mo}	**26.5**	0.12	**30.1**	0.23
	V_{mo}^*	8.2	0.08	17.2	0.21
$\lvert\mathcal{N}\rvert = 5$	V_1	28.3	0.21	32.6	0.25
	V_2	26.9	**0.03**	36.6	**0.15**
	V_{mo}	**26.3**	0.06	**26.9**	0.19
	V_{mo}^*	6.0	0.03	11.4	0.16

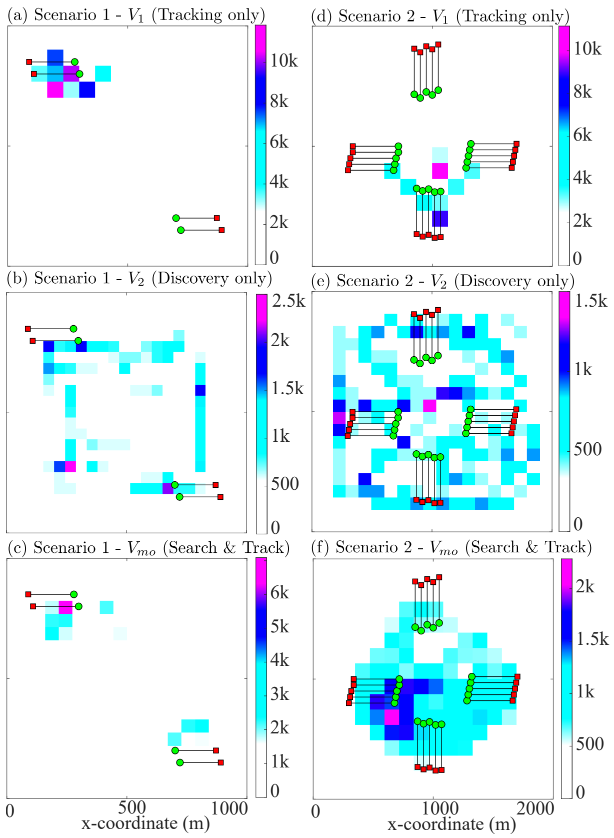

Fig. 10. Heatmap (16 × 16 grids) showing the positions of 5 UAVs and objects trajectories over 100 MC runs for Scenario 1 and Scenario 2. Circles (◯) and squares (□) indicate object start/stop positions. The colour legend represents UAV position frequency.

where $v \sim \mathcal{G}(\cdot; 0, R)$ is a 2D zero-mean Gaussian noise vector with covariance matrix $R = \text{diag}([\sigma_x^2, \sigma_y^2])$, $\sigma_x = \sigma_y = 5 + 0.01d(\boldsymbol{x}, u^{(n)})$, to model a distance-dependent uncertainty that increases as the object moves away from the agent. In addition to the detected measurements, the sensor also registers Poisson false alarms with rate $\lambda_c = 15$. The total simulation time is set to 200 s. All reported results are averaged over 100 Monte Carlo runs.

Table III summarises performance comparison for different value functions. The results indicate that *the proposed multi-objective value function*, V_{mo}, consistently outperforms the others in **OSPA**$^{(2)}$ tracking errors. This is because V_{mo} strategically balances between tracking already-detected objects and actively searching for new ones. The average grid occupancy entropy (measured in nats) sheds further insights into the coverage of the multi-agent team. A lower entropy value indicates a lower level of uncertainty in discovering new objects. As expected, V_2 exhibits the lowest entropy since it focuses solely on discovering, while V_{mo} balances between tracking and discovery, achieving an entropy higher than that of V_2 but lower than V_1.

Fig. 10 shows the trajectory heatmaps of five UAVs for both scenarios with different value functions. These results further substantiate the efficacy of our proposed multi-objective planning method. Specifically, for V_1 (tracking only; see Fig. 10(a) & (d)), the agents prioritise solely tracking detected objects and neglect any exploration tasks by only staying close to the detected objects in the northwestern region of Fig. 10(a), or in the southern region of Fig. 10(d). Conversely, V_2 (discovery only; see Fig. 10(b) & (e)) drives the agents to focus exclusively on exploration by circling the search area, disregarding tracking duties. In contrast, V_{mo} (joint search and track; see Fig. 10(c) & (f)) encourages the agents to track discovered objects and at the same time explore other parts of the surveillance area to discover new objects. For instance, in Fig. 10(c), the agents move from the northwestern region to the southeastern region to detect and track the two initially undetected objects. Similarly, in Fig. 10(f), the agents not only operate in the southern region but also venture north to track the five initially undetected objects.

V. CONCLUSIONS

We have developed a multi-objective planning method for a limited-FoV multi-agent team to discover and track multiple mobile objects with unknown data association. The problem is formulated as a multi-agent POMDP with the multi-object filtering density as the information state. An entropy-based multi-objective value function is developed and shown to be monotone and submodular, thereby enabling low-cost implementation via greedy search with tight optimality bound. A series of experiments on a real-world taxi dataset confirm our method's capability and efficiency. It also validates the robustness of our proposed multi-objective formulation, wherein its overall performance shows a decreasing trend over time similar to that of the best possible performance. So far, we have considered a centralised architecture for multi-agent planning where scalability can be a limitation. A scalable approach should be a distributed POMDP for MOT, where each agent runs its local filter to track multi-objects and coordinates with other agents to achieve a global objective. However, planning for multi-agents to reach a global goal under a distributed POMDP framework is an NEXP-complete problem [17].

REFERENCES

[1] H. V. Nguyen, M. Chesser, L. P. Koh, H. Rezatofighi, and D. C. Ranasinghe, "TrackerBots: Autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals," *Journal of Field Robotics*, vol. 36, no. 3, 2019.

[2] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House, 1999.

[3] N. Farmani, L. Sun, and D. Pack, "Optimal UAV sensor management and path planning for tracking multiple mobile targets," in *Proc. of DSCC*, vol. 2. American Society of Mechanical Engineers, 2014, pp. 1–8.

[4] S. Li, W. Yi, R. Hoseinnezhad, G. Battistelli, B. Wang, and L. Kong, "Robust distributed fusion with labeled random finite sets," *IEEE Trans. on Sig. Proc.*, vol. 66, no. 2, pp. 278–293, 2018.

[5] S. Li, G. Battistelli, L. Chisci, W. Yi, B. Wang, and L. Kong, "Computationally efficient multi-agent multi-object tracking with labeled random finite sets," *IEEE Trans. on Sig. Proc.*, vol. 67, no. 1, pp. 260–275, 2019.

[6] A. K. Gostar, T. Rathnayake, R. Tennakoon, A. Bab-Hadiashar, G. Battistelli, L. Chisci, and R. Hoseinnezhad, "Centralized cooperative sensor fusion for dynamic sensor network with limited field-of-view via labeled multi-Bernoulli filter," *IEEE Trans. on Sig. Proc.*, pp. 878–891, 2020.

[7] J. Ong, B.-T. Vo, B.-N. Vo, D. Y. Kim, and S. Nordholm, "A Bayesian filter for multi-view 3D multi-object tracking with occlusion handling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2246–2263, 2020.

[8] B. Wang, S. Li, G. Battistelli, L. Chisci, and W. Yi, "Multi-agent fusion with different limited fields-of-view," *IEEE Trans. on Sig. Proc.*, vol. 70, pp. 1560–1575, 2022.

[9] S. Panicker, A. K. Gostar, A. Bab-Hadiashar, and R. Hoseinnezhad, "Tracking of targets of interest using labeled multi-Bernoulli filter with multi-sensor control," *Signal Processing*, vol. 171, p. 107451, 2020.

[10] H. V. Nguyen, H. Rezatofighi, B.-N. Vo, and D. C. Ranasinghe, "Distributed multi-object tracking under limited field of view sensors," *IEEE Trans. on Sig. Proc.*, vol. 69, pp. 5329–5344, 2021.

[11] B. Wang, Y. Xu, S. Li, G. Battistelli, and L. Chisci, "Centralized multi-sensor labeled multi-Bernoulli filter with partially overlapping fields of view," *Signal Processing*, vol. 213, p. 109180, 2023.

[12] S. Welikala, C. G. Cassandras, H. Lin, and P. J. Antsaklis, "A new performance bound for submodular maximization problems and its application to multi-agent optimal coverage problems," *Automatica*, vol. 144, p. 110493, 2022.

[13] R. P. Mahler, *Advances in statistical multisource-multitarget information fusion*. Artech House, 2014.

[14] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[15] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *Trans. on Indus. Info.*, vol. 9, no. 1, pp. 132–141, 2012.

[16] J. B. Clempner and A. S. Poznyak, "Observer and control design in partially observable finite Markov chains," *Automatica*, vol. 110, p. 108587, 2019.

[17] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Math. of Op. Research*, vol. 27, no. 4, pp. 819–840, 2002.

[18] J. V. Messias, M. Spaan, and P. U. Lima, "Efficient offline communication policies for factored multiagent POMDPs," in *Proc. of 24th NeurIPS*, 2011, pp. 1917–1925.

[19] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The Inter. J. of Robotics Res.*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[20] X. Wang, R. Hoseinnezhad, A. K. Gostar, T. Rathnayake, B. Xu, and A. Bab-Hadiashar, "Multi-sensor control for multi-object Bayes filters," *Signal Processing*, vol. 142, pp. 260–270, 2018.

[21] B. Ristic and B.-N. Vo, "Sensor control for multi-object state-space estimation using random finite sets," *Automatica*, vol. 46, no. 11, pp. 1812–1818, 2010.

[22] H. G. Hoang and B. T. Vo, "Sensor management for multi-target tracking via multi-Bernoulli filtering," *Automatica*, vol. 50, pp. 1135–1142, 2014.

[23] Y. Zhu, J. Wang, and S. Liang, "Multi-objective optimization based multi-Bernoulli sensor selection for multi-target tracking," *Sensors*, vol. 19, no. 4, pp. 980 – 997, 2019.

[24] B. Vo, B. Vo, and M. Beard, "Multi-sensor multi-object tracking with the generalized labeled multi-Bernoulli filter," *IEEE Trans. on Sig. Proc.*, vol. 67, no. 23, pp. 5952–5967, Dec 2019.

[25] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[26] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Trans. on Sig. Proc.*, vol. 62, no. 12, pp. 3246–3260, 2014.

[27] [dataset] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset EPFL/mobility (v. 2009-02-24)," Feb. 2009.

[28] H. V. Nguyen, H. Rezatofighi, B.-N. Vo, and D. C. Ranasinghe, "Multi-objective multi-agent planning for jointly discovering and tracking mobile objects," in *Proc. of 34th AAAI*, Feb 2020, pp. 7227–7235.

[29] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. on Sig. Proc.*, vol. 61, no. 13, pp. 3460–3475, 2013.

[30] B.-N. Vo and B.-T. Vo, "A multi-scan labeled random finite set model for multi-object state estimation," *IEEE Trans. on Sig. Proc.*, vol. 67, no. 19, pp. 4948–4963, 2019.

[31] M. Corah and N. Michael, "Distributed submodular maximization on partition matroids for planning on large sensor networks," in *Proc. of CDC*. IEEE, 2018, pp. 6792–6799.

[32] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, 2007.

[33] W. Liu, S. Zhu, C. Wen, and Y. Yu, "Structure modeling and estimation of multiple resolvable group targets via graph theory and multi-Bernoulli filter," *Automatica*, vol. 89, 2018.

[34] M. Hauskrecht, "Value-function approximations for partially observable Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 13, pp. 33–94, 2000.

[35] F. Chen, H. V. Nguyen, D. A. Taggart, K. Falkner, S. H. Rezatofighi, and D. C. Ranasinghe, "ConservationBots: Autonomous aerial robot for fast robust wildlife tracking in complex terrains," *Journal of Field Robotics*, vol. 41, no. 2, pp. 443–469, 2024.

[36] R. Mahler, "A GLMB filter for unified multitarget multisensor management," in *Proc of SPIE*, vol. 11018, 2019, p. 110180D.

[37] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.

[38] D. Driess, D. Hennes, and M. Toussaint, "Active multi-contact continuous tactile exploration with Gaussian process differential entropy," in *Proc. of ICRA*. IEEE, 2019, pp. 7844–7850.

[39] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. on Aero. and Elect. Sys.*, vol. 41, no. 4, pp. 1224–1245, 2005.

[40] J. Koski, "Multicriterion structural optimization," in *Optimization of Large Structural Systems*. Springer Netherlands, 1993, pp. 793–809.

[41] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[42] G. Qu, D. Brown, and N. Li, "Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions," *Automatica*, vol. 105, pp. 206–215, 2019.

[43] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.

[44] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, no. 93, pp. 2761–2801, 2008.

[45] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Prog.*, vol. 14, no. 1, pp. 265–294, 1978.

[46] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—II," in *Poly. Comb.* Springer, 1978, pp. 73–87.

[47] G. Guido, V. Gallelli, D. Rogano, and A. Vitale, "Evaluating the accuracy of vehicle tracking data obtained from unmanned aerial vehicles," *Inter. J. of Transport. Sci. and Tech.*, vol. 5, no. 3, pp. 136–151, 2016.

[48] J. Zhu, K. Sun, S. Jia, Q. Li, X. Hou, W. Lin, B. Liu, and G. Qiu, "Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 4968–4981, 2018.

[49] M. Beard, B. T. Vo, and B.-N. Vo, "A solution for large-scale multi-object tracking," *IEEE Trans. on Sig. Proc.*, vol. 68, pp. 2754–2769, 2020.

[50] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Proc. of ICRA*. IEEE, 2010, pp. 4791–4796.

[51] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Proc. of RSS*, 2015.

[52] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2009.

[53] R. A. MacDonald and S. L. Smith, "Active sensing for motion planning in uncertain environments via mutual information policies," *The Int. Journal of Robotics Research*, vol. 38, no. 2-3, pp. 146–161, 2019.

[54] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Trans. on Mechatronics*, vol. 18, no. 6, pp. 1827–1838, 2012.

[55] K. Shen, C. Zhang, P. Dong, Z. Jing, and H. Leung, "Consensus-based labeled multi-Bernoulli filter with event-triggered communication," *IEEE Trans. on Sig. Proc.*, vol. 70, pp. 1185–1196, 2022.

Appendix

A. MS-GLMB Recursion

Within the conventional multi-object framework, the MS-GLMB recursion, denoted by Ω, propagates the GLMB parameter set:

$$\boldsymbol{\pi} \triangleq \left\{ \left(w^{(\xi,I)}, p^{(\xi)} \right) : (\xi, I) \in \Xi \times \mathcal{F}(\mathbb{L}) \right\},$$

to the parameter set [24]

$$\boldsymbol{\pi}_+ = \left\{ \left(w_+^{(\xi_+,I_+)}, p_+^{(\xi_+)} \right) : (\xi_+, I_+) \in \Xi_+ \times \mathcal{F}(\mathbb{L}_+) \right\},$$

where:

$$I_+ = I \uplus \mathbb{B}_+,\ \xi_+ = (\xi, \gamma_+), \tag{30}$$

$$w_+^{(I_+,\xi_+)} = 1_{\mathcal{F}(\mathbb{B}_+ \uplus I)}(\mathcal{L}(\gamma_+)) w^{(I,\xi)} \left[\omega^{(\xi,\gamma_+)} \right]^{\mathbb{B}_+ \uplus I}, \tag{31}$$

$$p_+^{(\xi_+)}(x_+, \ell) \propto \begin{cases} \langle p^{(\xi)}(\cdot, \ell), \Upsilon_S^{(\gamma_+)}(x_+|\cdot, \ell) \rangle, & \ell \in \mathcal{L}(\gamma_+) \setminus \mathbb{B}_+ \\ \Upsilon_B^{(\gamma_+)}(x_+, \ell), & \ell \in \mathcal{L}(\gamma_+) \cap \mathbb{B}_+ \end{cases}, \tag{32}$$

$$\omega^{(\xi,\gamma_+)}(\ell) = \begin{cases} 1 - \bar{r}_S^{(\xi)}(\ell), & \ell \in \overline{\mathcal{L}(\gamma_+)} \setminus \mathbb{B}_+ \\ \overline{\Upsilon}_S^{(\xi,\gamma_+)}(\ell), & \ell \in \mathcal{L}(\gamma_+) \setminus \mathbb{B}_+ \\ 1 - r_{B,+}(\ell), & \ell \in \overline{\mathcal{L}(\gamma_+)} \cap \mathbb{B}_+ \\ \overline{\Upsilon}_B^{(\gamma_+)}(\ell), & \ell \in \mathcal{L}(\gamma_+) \cap \mathbb{B}_+ \end{cases}, \tag{33}$$

$$\bar{r}_S^{(\xi)}(\ell) = \langle p^{(\xi)}(\cdot, \ell), r_S(\cdot, \ell) \rangle, \tag{34}$$

$$\Upsilon_B^{(\gamma_+)}(x_+, \ell) = p_{B,+}(x_+, \ell) r_{B,+}(\ell) \psi^{(\gamma_+)}(x_+, \ell), \tag{35}$$

$$\Upsilon_S^{(\gamma_+)}(x_+|y, \ell) = f_+(x_+|y, \ell) r_S(y, \ell) \psi^{(\gamma_+)}(x_+, \ell), \tag{36}$$

$$\overline{\Upsilon}_B^{(\gamma_+)}(\ell) = \int \Upsilon_B^{(\gamma_+)}(x, \ell) dx, \tag{37}$$

$$\overline{\Upsilon}_S^{(\xi,\gamma_+)}(\ell) = \int \langle p^{(\xi)}(\cdot, \ell), \Upsilon_S^{(\gamma_+)}(x|\cdot, \ell) \rangle dx, \tag{38}$$

$$\psi^{(\gamma)}(\boldsymbol{x}) \triangleq \prod_{n \in \mathcal{N}} \psi^{(\gamma^{(n)},n)}(\boldsymbol{x}), \tag{39}$$

$$\psi^{(\gamma^{(n)},n)}(\boldsymbol{x}) = \begin{cases} 1 - P_D^{(n)}(\boldsymbol{x}), & \gamma^{(n)}(\mathcal{L}(\boldsymbol{x})) = 0 \\ \frac{P_D^{(n)}(\boldsymbol{x}) g^{(n)}(z_j^{(n)}|\boldsymbol{x})}{\kappa^{(a)}(z_j^{(n)})}, & \gamma^{(n)}(\mathcal{L}(\boldsymbol{x})) = j > 0 \end{cases}.$$

B. Analytical Form of Tracking Value Function

To prove Proposition **1**, we first require the three following Lemmas.

Lemma 1: For $f : \mathcal{F}(\mathbb{L}) \to \mathbb{R}$, and $g : \mathbb{L} \to \mathbb{R}$,:

$$\sum_{L \subseteq \mathbb{L}} f(L) \sum_{\ell \in L} g(\ell) = \sum_{\ell \in \mathbb{L}} g(\ell) \sum_{L \subseteq \mathbb{L} \setminus \{\ell\}} f(L \cup \{\ell\}) \tag{40}$$

Proof: Notably, a specific case ($f(L) = [1 - r_1^{(\ell)}]^{\mathbb{L} \setminus L} [r_1^{(\ell)}]^L$ and $g(\ell) = \ln(r_1^{(\ell)}/r_2^{(\ell)})$) of this result was first used in Proposition 1 of [55] without proof. This lemma presents a more general result and is proven via induction.

It is trivial to confirm that (40) holds for $\mathbb{L} = \emptyset$ and $\mathbb{L} = \{\ell\}$. Suppose (40) holds for $\mathbb{L} = \{\ell_1, \ldots, \ell_n\}$. It remains to show that (40) also holds for $\mathbb{L} \cup \{\hat{\ell}\}$, i.e.,

$$\sum_{L \subseteq \mathbb{L} \cup \hat{\ell}} f(L) \sum_{\ell \in L} g(\ell) = \sum_{\ell \in \mathbb{L} \cup \hat{\ell}} g(\ell) \sum_{L \subseteq \mathbb{L} \cup \hat{\ell} \setminus \{\ell\}} f(L \cup \{\ell\}). \tag{41}$$

Note that for any function $f : \mathcal{F}(\mathbb{L}) \to \mathbb{R}$,

$$\sum_{L \subseteq \mathbb{L} \cup \hat{\ell}} f(L) = \sum_{L \subseteq \mathbb{L}} f(L) + \sum_{L \subseteq \mathbb{L}} f(L \cup \{\hat{\ell}\}) \tag{42}$$

since $\mathcal{F}(\mathbb{L} \cup \hat{\ell}) = \mathcal{F}(\mathbb{L}) \cup \{(L \cup \{\hat{\ell}\}) : L \in \mathcal{F}(\mathbb{L})\}$ where $\mathcal{F}(\cdot)$ denotes the class of finite subsets. Hence,

$$\begin{aligned} &\sum_{L \subseteq \mathbb{L} \cup \hat{\ell}} f(L) \sum_{\ell \in L} g(\ell) \\ &= \sum_{L \subseteq \mathbb{L}} f(L) \sum_{\ell \in L} g(\ell) + \sum_{L \subseteq \mathbb{L}} f(L \cup \{\hat{\ell}\}) \sum_{\ell \in L \cup \{\hat{\ell}\}} g(\ell) \end{aligned} \tag{43}$$

Substituting (40) into (43), and noting that $\sum_{\ell \in L \cup \{\hat{\ell}\}} g(\ell) = \sum_{\ell \in L} g(\ell) + g(\hat{\ell})$, we have:

$$\begin{aligned} &\sum_{L \subseteq \mathbb{L} \cup \hat{\ell}} f(L) \sum_{\ell \in L} g(\ell) \\ &= \sum_{\ell \in \mathbb{L}} g(\ell) \sum_{L \subseteq \mathbb{L} \setminus \{\ell\}} f(L \cup \{\ell\}) + \sum_{L \subseteq \mathbb{L}} f(L \cup \{\hat{\ell}\}) \sum_{\ell \in L} g(\ell) \\ &\quad + \sum_{L \subseteq \mathbb{L}} f(L \cup \{\hat{\ell}\}) g(\hat{\ell}) \\ &= \sum_{\ell \in \mathbb{L}} g(\ell) \sum_{L \subseteq \mathbb{L} \setminus \{\ell\}} f(L \cup \{\ell\}) \\ &\quad + \sum_{\ell \in \mathbb{L}} g(\ell) \sum_{L \subseteq \mathbb{L} \setminus \{\ell\}} f(L \cup \{\hat{\ell}\} \cup \{\ell\}) + g(\hat{\ell}) \sum_{L \subseteq \mathbb{L}} f(L \cup \{\hat{\ell}\}) \\ &= \sum_{\ell \in \mathbb{L}} g(\ell) \sum_{L \subseteq \mathbb{L} \cup \hat{\ell} \setminus \{\ell\}} f(L \cup \{\ell\}) + g(\hat{\ell}) \sum_{L \subseteq \mathbb{L}} f(L \cup \{\hat{\ell}\}) \\ &= \sum_{\ell \in \mathbb{L} \cup \hat{\ell}} g(\ell) \sum_{L \subseteq \mathbb{L} \cup \hat{\ell} \setminus \{\ell\}} f(L \cup \{\ell\}) \quad \square. \end{aligned}$$

Lemma 2: For $f : \mathcal{F}(\mathbb{L}) \to \mathbb{R}$, and $g : \mathbb{L} \to \mathbb{R}$, we have:

$$\sum_{L \subseteq \mathbb{L}} f(L) \sum_{\ell \in \mathbb{L} \setminus L} g(\ell) = \sum_{\ell \in \mathbb{L}} g(\ell) \sum_{L \subseteq \mathbb{L} \setminus \{\ell\}} f(L). \tag{44}$$

Proof: We prove this Lemma via induction. It is clear that (44) holds for $\mathbb{L} = \emptyset$ and $\mathbb{L} = \{\ell\}$. Now, assuming (44) for $\mathbb{L} = \{\ell_1, \ldots, \ell_n\}$, we demonstrate its validity for $\mathbb{L} \cup \hat{\ell}$, i.e.,:

$$\sum_{L \subseteq \mathbb{L} \cup \hat{\ell}} f(L) \sum_{\ell \in \mathbb{L} \cup \hat{\ell} \setminus L} g(\ell) = \sum_{\ell \in \mathbb{L} \cup \hat{\ell}} g(\ell) \sum_{L \subseteq \mathbb{L} \cup \hat{\ell} \setminus \{\ell\}} f(L). \tag{45}$$

Using (42) and noting that $\sum_{\ell \in \mathbb{L} \cup \{\hat{\ell}\} \setminus L} g(\ell) = \sum_{\ell \in \mathbb{L} \setminus L} g(\ell) + g(\hat{\ell})$, we have:

$$\begin{aligned} &\sum_{L \subseteq \mathbb{L} \cup \hat{\ell}} f(L) \sum_{\ell \in \mathbb{L} \cup \hat{\ell} \setminus L} g(\ell) \\ &= \sum_{L \subseteq \mathbb{L}} f(L) \sum_{\ell \in \mathbb{L} \cup \hat{\ell} \setminus L} g(\ell) + \sum_{L \subseteq \mathbb{L}} f(L \cup \hat{\ell}) \sum_{\ell \in \mathbb{L} \setminus L} g(\ell) \\ &= \sum_{L \subseteq \mathbb{L}} f(L) \sum_{\ell \in \mathbb{L} \setminus L} g(\ell) + \sum_{L \subseteq \mathbb{L}} f(L) g(\hat{\ell}) + \sum_{L \subseteq \mathbb{L}} f(L \cup \hat{\ell}) \sum_{\ell \in \mathbb{L} \setminus L} g(\ell). \end{aligned} \tag{46}$$

Substituting (44) into (46), we have:

$$\begin{aligned}
&\sum_{L\subseteq\mathbb{L}\cup\hat{\ell}} f(L) \sum_{\ell\in\mathbb{L}\cup\hat{\ell}\setminus L} g(\ell)\\
&=\sum_{\ell\in\mathbb{L}} g(\ell) \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} f(L) + g(\hat{\ell})\sum_{L\subseteq\mathbb{L}} f(L) + \sum_{\ell\in\mathbb{L}} g(\ell) \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} f(L\cup\hat{\ell})\\
&=\sum_{\ell\in\mathbb{L}} g(\ell) \sum_{L\subseteq\mathbb{L}\cup\hat{\ell}\setminus\{\ell\}} f(L) + g(\hat{\ell}) \sum_{L\subseteq\mathbb{L}\cup\hat{\ell}\setminus\{\ell\}} f(L)\\
&=\sum_{\ell\in\mathbb{L}\cup\hat{\ell}} g(\ell) \sum_{L\subseteq\mathbb{L}\cup\hat{\ell}\setminus\{\ell\}} f(L) \quad \square.
\end{aligned}$$

Lemma 3: For $f:\mathcal{F}(\mathbb{L})\to\mathbb{R}$, $p:\mathbb{X}\times\mathbb{L}\to\mathbb{R}$, $q:\mathbb{X}\times\mathbb{L}\to\mathbb{R}$ with q is a unitless function, we have:

$$\int \Delta(\boldsymbol{X}) f(\mathcal{L}(\boldsymbol{X})) p^{\boldsymbol{X}} \ln q^{\boldsymbol{X}} \delta\boldsymbol{X} = \sum_{L\subseteq\mathbb{L}} f(L) \langle p\rangle^{L} \sum_{\ell\in L} \frac{\langle p\ln q\rangle(\ell)}{\langle p\rangle(\ell)}.$$

Proof: We first note that

$$\begin{aligned}
I &\triangleq \int \prod_{j=1}^{i} p(x_j,\ell_j) \ln\left(\prod_{n=1}^{i} q(x_n,\ell_n)\right) dx_{1:i}\\
&= \int \prod_{j=1}^{i} p(x_j,\ell_j) \sum_{n=1}^{i} \ln q(x_n,\ell_n) dx_{1:i}\\
&= \sum_{n=1}^{i} \int p(x,\ell_n)\ln q(x,\ell_n)dx \prod_{j\in\{1,\ldots,i\}\setminus\{n\}} \int p(x,\ell_j)dx\\
&= \sum_{\ell\in\{\ell_{1:i}\}} \langle p\ln q\rangle(\ell)\langle p\rangle^{\{\ell_{1:i}\}-\{\ell\}}. \qquad (47)
\end{aligned}$$

Now

$$\begin{aligned}
&\int \Delta(\boldsymbol{X}) f(\mathcal{L}(\boldsymbol{X})) p^{\boldsymbol{X}} \ln q^{\boldsymbol{X}} \delta\boldsymbol{X}\\
&= f(\emptyset)p^{\emptyset}\ln q^{\emptyset} + \sum_{i=1}^{\infty}\frac{1}{i!}\sum_{\ell_{1:i}} \delta_i[|\{\ell_{1:i}\}|] f(\{\ell_1,\ldots,\ell_i\}) \times I,\\
&= \sum_{i=1}^{\infty}\frac{1}{i!}\sum_{\ell_{1:i}} \delta_i[|\{\ell_{1:i}\}|] f(\{\ell_1,\ldots,\ell_i\}) \times I, \qquad (48)\\
&= \sum_{i=1}^{\infty}\frac{1}{i!} i! \sum_{L:|L|=i} f(L)\left[\sum_{\ell\in L}\langle p\rangle^{L-\{\ell\}}\langle p\ln q\rangle(\ell)\right] \qquad (49)\\
&= \sum_{L\subseteq\mathbb{L}} f(L)\langle p\rangle^{L}\sum_{\ell\in L}\frac{\langle p\ln q\rangle(\ell)}{\langle p\rangle(\ell)} \qquad (50)
\end{aligned}$$

where (49) follows by substituting (47) into (48), and the last step follows from Lemma 12 in [29], and noting that

$$f(\emptyset)\langle p\rangle^{\emptyset}\sum_{\ell\in\emptyset}\frac{\langle p\ln q\rangle(\ell)}{\langle p\rangle(\ell)} = 0 \quad \square. \qquad (51)$$

Proof of Proposition 1: For the LMB density, we have $\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}}$; hence:

$$\begin{aligned}
-h(\boldsymbol{X}) &= \int \boldsymbol{\pi}(\boldsymbol{X})\ln\left(K^{|\boldsymbol{X}|}\boldsymbol{\pi}(\boldsymbol{X})\right)\delta\boldsymbol{X}\\
&= \int \Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}} \ln\left[\Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}} K^{|\boldsymbol{X}|}\right]\delta\boldsymbol{X}\\
&= \int \Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}} \ln\left[\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})}\right]\delta\boldsymbol{X}\\
&\quad + \int \Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}} \ln\left[p^{\boldsymbol{X}} K^{|\boldsymbol{X}|}\right]\delta\boldsymbol{X}. \qquad (52)
\end{aligned}$$

The first term on the right-hand side (RHS) of (52) is:

$$\begin{aligned}
&\int \Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}} \ln\left[\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})}\right]\delta\boldsymbol{X} \qquad (53)\\
&= \sum_{L\subseteq\mathbb{L}} \tilde{r}^{\mathbb{L}\setminus L} r^{L} \ln\left[\tilde{r}^{\mathbb{L}\setminus L} r^{L}\right]\left[\int p(x,\cdot)dx\right]^{L} \qquad (54)\\
&= \sum_{L\subseteq\mathbb{L}} \tilde{r}^{\mathbb{L}\setminus L} r^{L} \sum_{\ell\in\mathbb{L}\setminus L}\ln\tilde{r}^{(\ell)} + \sum_{L\subseteq\mathbb{L}} \tilde{r}^{\mathbb{L}\setminus L} r^{L}\sum_{\ell\in L}\ln r^{(\ell)} \qquad (55)\\
&= \sum_{\ell\in\mathbb{L}} \ln\tilde{r}^{(\ell)} \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{\mathbb{L}\setminus L} r^{L}\\
&\quad + \sum_{\ell\in\mathbb{L}} \ln r^{(\ell)} \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{\mathbb{L}\setminus(L\cup\{\ell\})} r^{L\cup\{\ell\}} \qquad (56)\\
&= \sum_{\ell\in\mathbb{L}} \Bigg[\tilde{r}^{(\ell)}\ln\tilde{r}^{(\ell)} \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{(\mathbb{L}\setminus\{\ell\})\setminus L} r^{L}\\
&\quad + r^{(\ell)}\ln r^{(\ell)} \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{(\mathbb{L}\setminus\{\ell\})\setminus L} r^{L}\Bigg] \qquad (57)\\
&= \sum_{\ell\in\mathbb{L}} \left[r^{(\ell)}\ln r^{(\ell)} + \tilde{r}^{(\ell)}\ln\tilde{r}^{(\ell)}\right]. \qquad (58)
\end{aligned}$$

Here, (54) follows from Lemma 3 in [29]; (55) follows from $\int p(x,\cdot)dx = 1$; each term in (56) follows from Lemma 2 and Lemma 1, respectively; while (58) follows from the Binomial Theorem, i.e., $\sum_{L\subseteq\mathbb{L}} f^{\mathbb{L}\setminus L} g^{L} = (f+g)^{\mathbb{L}}$. Hence, $\sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{(\mathbb{L}\setminus\{\ell\})\setminus L} r^{L} = (\tilde{r}+r)^{\mathbb{L}\setminus\{\ell\}} = 1$.

The second term of the RHS of (52) is:

$$\begin{aligned}
&\int \Delta(\boldsymbol{X})\tilde{r}^{\mathbb{L}\setminus\mathcal{L}(\boldsymbol{X})} r^{\mathcal{L}(\boldsymbol{X})} p^{\boldsymbol{X}} \ln\left[p^{\boldsymbol{X}} K^{|\boldsymbol{X}|}\right]\delta\boldsymbol{X} \qquad (59)\\
&= \sum_{L\subseteq\mathbb{L}} \tilde{r}^{\mathbb{L}\setminus L} r^{L} \sum_{\ell\in L}\left\langle p^{(\ell)}\ln(Kp^{(\ell)})\right\rangle \qquad (60)\\
&= \sum_{\ell\in\mathbb{L}} \left\langle p^{(\ell)}\ln(Kp^{(\ell)})\right\rangle \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{\mathbb{L}\setminus(L\cup\{\ell\})} r^{L\cup\{\ell\}} \qquad (61)\\
&= \sum_{\ell\in\mathbb{L}} r^{(\ell)} \left\langle p^{(\ell)}\ln(Kp^{(\ell)})\right\rangle \sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{(\mathbb{L}\setminus\{\ell\})\setminus L} r^{L} \qquad (62)\\
&= \sum_{\ell\in\mathbb{L}} r^{(\ell)} \left\langle p^{(\ell)}\ln(Kp^{(\ell)})\right\rangle. \qquad (63)
\end{aligned}$$

Here, (60) follows from Lemma 3; (61) follows from Lemma 1; while (63) follows from the Binomial Theorem, i.e., $\sum_{L\subseteq\mathbb{L}\setminus\{\ell\}} \tilde{r}^{(\mathbb{L}\setminus\{\ell\})\setminus L} r^{L} = (\tilde{r}+r)^{\mathbb{L}\setminus\{\ell\}} = 1$.

Substituting (58) and (63) into (52), we have:

$$h(\boldsymbol{X}) = -\sum_{\ell\in\mathbb{L}}\left[r^{(\ell)}\ln r^{(\ell)} + \tilde{r}^{(\ell)}\ln\tilde{r}^{(\ell)} + r^{(\ell)}\left\langle p^{(\ell)}\ln(Kp^{(\ell)})\right\rangle\right] \square.$$

C. Analytical Form of Discovery Value Function

Proof of Proposition 2: The predicted occupancy probability in (18) is straight forward. To establish (19), note that

$$\begin{aligned}
\Pr(O_{j+1}^{(i)}=1, Y_{a_j,j+1}^{(i)}=1) &= \omega_{j+1}^{(i)}(a_{j-1})Q_{D,+}^{(i)}(a_j),\\
\Pr(O_{j+1}^{(i)}=0, Y_{a_j,j+1}^{(i)}=1) &= 1-\omega_{j+1}^{(i)}(a_{j-1}).
\end{aligned}$$

Therefore, $\Pr(Y_{a_j,j+1}^{(i)}=1) = 1-\omega_{j+1}^{(i)}(a_{j-1}) + \omega_{j+1}^{(i)}(a_{j-1})Q_{D,+}^{(i)}(a_j)$, and using Bayes' rule yields (19).

Note that if $\varkappa^{(i)}$ generates measurements (i.e., $Y_{a,j}^{(i)} = 0$), then it is occupied (i.e., $O_j^{(i)} = 1$), which means

$$\begin{aligned}\Pr(O_j^{(i)} = 0|Y_{a,j}^{(i)} = 0) &= 0.\\ \Pr(O_j^{(i)} = 1|Y_{a,j}^{(i)} = 0) &= 1.\end{aligned}$$

Hence, $h(O_j^{(i)}|Y_{a,j}^{(i)} = 0) = 0\ln 0 + 1\ln 1 = 0$, and

$$h(O_j^{(i)}|Y_{a,j}^{(i)}) = \Pr(Y_{a,j}^{(i)} = 1)h(O_j^{(i)}|Y_{a,j}^{(i)} = 1). \tag{64}$$

Substituting for $\Pr(O_j^{(i)} = 1|Y_{a,j}^{(i)} = 1)$ and $\Pr(Y_{a,j}^{(i)} = 1)$ yields (17) □.

D. Monotone Submodularity

To prove Proposition **3**, we first require the following result.

Lemma 4: $\forall \boldsymbol{R} \subseteq \boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$ and $\forall \boldsymbol{Y} \in \boldsymbol{\mathcal{Z}} \setminus \boldsymbol{Z}$:

$$I(\boldsymbol{X};\boldsymbol{Z},\boldsymbol{Y}) - I(\boldsymbol{X};\boldsymbol{Z}) \leq I(\boldsymbol{X};\boldsymbol{R},\boldsymbol{Y}) - I(\boldsymbol{X};\boldsymbol{R}).$$

Proof : Since $\boldsymbol{R} \subseteq \boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$, using the mutual information inequalities [25, p.50], we have:

$$\begin{aligned}& I(\boldsymbol{Z};\boldsymbol{Y}) \geq I(\boldsymbol{R};\boldsymbol{Y})\\ \Leftrightarrow\ & h(\boldsymbol{Y}) - h(\boldsymbol{Y}|\boldsymbol{Z}) \geq h(\boldsymbol{Y}) - h(\boldsymbol{Y}|\boldsymbol{R}) \quad (65)\\ \Leftrightarrow\ & h(\boldsymbol{Y}|\boldsymbol{R}) \geq h(\boldsymbol{Y}|\boldsymbol{Z}),\\ \Leftrightarrow\ & h(\boldsymbol{R},\boldsymbol{Y}) - h(\boldsymbol{R}) \geq h(\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{Z}). \quad (66)\end{aligned}$$

Further, nothing that $I(\boldsymbol{Z};\boldsymbol{Y}|\boldsymbol{X}) = I(\boldsymbol{R};\boldsymbol{Y}|\boldsymbol{X}) = 0$ because $\boldsymbol{Y}$ is independent of $\boldsymbol{R}$ and $\boldsymbol{Z}$, given $\boldsymbol{X}$, we have,

$$\begin{aligned}h(\boldsymbol{Y}|\boldsymbol{X}) &= h(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{Z}) + I(\boldsymbol{Z};\boldsymbol{Y}|\boldsymbol{X}) = h(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{Z})\\ &= h(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{X},\boldsymbol{Z}),\\ &= h(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{R}) + I(\boldsymbol{R};\boldsymbol{Y}|\boldsymbol{X})\\ &= h(\boldsymbol{X},\boldsymbol{R},\boldsymbol{Y}) - h(\boldsymbol{X},\boldsymbol{R}).\end{aligned}$$

and hence,

$$h(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{X},\boldsymbol{Z}) = h(\boldsymbol{X},\boldsymbol{R},\boldsymbol{Y}) - h(\boldsymbol{X},\boldsymbol{R}). \tag{67}$$

Subtracting (66) from (67), we have:

$$\begin{aligned}&[h(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{X},\boldsymbol{Z})] - [h(\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{Z})]\\ &\quad \geq [h(\boldsymbol{X},\boldsymbol{R},\boldsymbol{Y}) - h(\boldsymbol{X},\boldsymbol{R})] - [h(\boldsymbol{R},\boldsymbol{Y}) - h(\boldsymbol{R})]\end{aligned}$$

Note from the differential entropy chain rules [25, p.253] that $h(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{Y}) = h(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{Z},\boldsymbol{Y})$ and $h(\boldsymbol{X}|\boldsymbol{Z}) = h(\boldsymbol{X},\boldsymbol{Z}) - h(\boldsymbol{Z})$. Substituting these into the above gives

$$\begin{aligned}& h(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{Y}) - h(\boldsymbol{X}|\boldsymbol{Z}) \geq h(\boldsymbol{X}|\boldsymbol{R},\boldsymbol{Y}) - h(\boldsymbol{X}|\boldsymbol{R})\\ \Leftrightarrow & [h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{Y})] - [h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{Z})]\\ & \quad \leq [h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{R},\boldsymbol{Y}] - [h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{R})],\\ \Leftrightarrow & I(\boldsymbol{X};\boldsymbol{Z},\boldsymbol{Y}) - I(\boldsymbol{X};\boldsymbol{Z}) \leq I(\boldsymbol{X};\boldsymbol{R},\boldsymbol{Y}) - I(\boldsymbol{X};\boldsymbol{R}).\Box\end{aligned}$$

Proof of Proposition 3: Denote $\breve{\boldsymbol{\pi}}(\boldsymbol{Z}_1 \cup \boldsymbol{Z}_2) \equiv \boldsymbol{\pi}(\boldsymbol{Z}_1,\boldsymbol{Z}_2)$, then according to (3.53) and (25.36) in [13]: $\int \breve{\boldsymbol{\pi}}(\boldsymbol{Z}_1 \cup \boldsymbol{Z}_2)\delta(\boldsymbol{Z}_1 \cup \boldsymbol{Z}_2) = \int \boldsymbol{\pi}(\boldsymbol{Z}_1,\boldsymbol{Z}_2)\delta\boldsymbol{Z}_1\delta\boldsymbol{Z}_2$. Further, noting that

$$I(\boldsymbol{X};\boldsymbol{Z}) = \int \boldsymbol{\pi}(\boldsymbol{X},\boldsymbol{Z})\ln\left(\frac{\boldsymbol{\pi}(\boldsymbol{X},\boldsymbol{Z})}{\boldsymbol{\pi}(\boldsymbol{X})\boldsymbol{\pi}(\boldsymbol{Z})}\right)\delta\boldsymbol{X}\delta\boldsymbol{Z}$$

$$I(\boldsymbol{X};\boldsymbol{Z_1},\boldsymbol{Z_2}) = \int \boldsymbol{\pi}(\boldsymbol{X},\boldsymbol{Z_1},\boldsymbol{Z_2})\ln\left(\frac{\boldsymbol{\pi}(\boldsymbol{X},\boldsymbol{Z_1},\boldsymbol{Z_2})}{\boldsymbol{\pi}(\boldsymbol{X})\boldsymbol{\pi}(\boldsymbol{Z_1},\boldsymbol{Z_2})}\right)\delta\boldsymbol{X}\delta\boldsymbol{Z_1}\delta\boldsymbol{Z_2}.$$

we have $I(\boldsymbol{X};\boldsymbol{Z}_1 \cup \boldsymbol{Z}_2) = I(\boldsymbol{X};\boldsymbol{Z}_1,\boldsymbol{Z}_2)$.

Due to the mutual exclusiveness of the multi-object measurements between the agents, and the independence of an agent's measurement from the other agent's actions, $\boldsymbol{Z}_{\boldsymbol{A},j} = \uplus_{\boldsymbol{a}\in\boldsymbol{A}}\boldsymbol{Z}_{\boldsymbol{a},j}$, where $\boldsymbol{Z}_{(a^{(m)},m),j}$ is the measurement set received by agent m at time j after it has taken action $a^{(m)}$. Hence, for $\boldsymbol{A} \subseteq \boldsymbol{B} \subset \mathbb{A}$, $\boldsymbol{a} \in \mathbb{A} \setminus \boldsymbol{B}$, we have: $\boldsymbol{Z}_{\boldsymbol{A}} \subseteq \boldsymbol{Z}_{\boldsymbol{B}}$ and $\boldsymbol{Z}_{\boldsymbol{a}} \in \boldsymbol{\mathcal{Z}} \setminus \boldsymbol{Z}_{\boldsymbol{B}}$. Thus, it follows from Lemma 4 that

$$I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{B}},\boldsymbol{Z}_{\boldsymbol{a}}) - I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{B}}) \leq I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}},\boldsymbol{Z}_{\boldsymbol{a}}) - I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}}),$$

which is equivalent to

$$I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{B}\cup\{\boldsymbol{a}\}}) - I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{B}}) \leq I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}\cup\{\boldsymbol{a}\}} - I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}}).$$

i.e., $I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}})$ is a *submodular* set function. Further, using the chain rule we have:

$$I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}},\boldsymbol{Z}_{\boldsymbol{a}}) - I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}}) = I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}}|\boldsymbol{Z}_{\boldsymbol{a}}) \geq 0$$

i.e., $I(\boldsymbol{X};\boldsymbol{Z}_{\boldsymbol{A}})$ is a *monotone submodular* set function □.

Proof of Proposition 5: Let $\boldsymbol{Y}_{\boldsymbol{A}}^{(i)} = \uplus_{\boldsymbol{a}\in\boldsymbol{A}}\boldsymbol{Y}_{\boldsymbol{a}}^{(i)}$, where $\boldsymbol{Y}_{(a^{(m)},m)}^{(i)} = (Y_{(a^{(m)},m)}^{(i)}, m)$, and $Y_{(a^{(m)},m)}^{(i)}$ is the binary measurement $\delta_{\emptyset}[\boldsymbol{Z}_{(a^{(m)},m)}(\varkappa^{(i)})]$ received by agent m from cell $\varkappa^{(i)}$, after it has taken action $(a^{(m)}, m)$. Note that the binary measurement $Y_{\boldsymbol{A}}^{(i)} = \delta_{\emptyset}[\boldsymbol{Z}_{\boldsymbol{A}}(\varkappa^{(i)})]$ received from cell $\varkappa^{(i)}$ can be written as $\prod_{\boldsymbol{a}\in\boldsymbol{A}} Y_{\boldsymbol{a}}^{(i)}$, due to the independence of the agent's measurement from the other agent's actions. We first show by induction that

$$h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)}) = h(O^{(i)}|Y_{\boldsymbol{A}}^{(i)}) \tag{68}$$

It is trivial to confirm that (68) holds for $\{\boldsymbol{a}\}$. Suppose that (68) holds for a non-empty $\boldsymbol{A}$, then

$$h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)}) = h(O^{(i)}|Y_{\boldsymbol{A}}^{(i)}) = \Pr(Y_{\boldsymbol{A}}^{(i)} = 1)h(O^{(i)}|Y_{\boldsymbol{A}}^{(i)} = 1). \tag{69}$$

Decomposing $\boldsymbol{Y}_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)}$, we have

$$\begin{aligned}h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)}) &= h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)},\boldsymbol{Y}_{\boldsymbol{a}}^{(i)})\\ &= \Pr(\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 0)h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)},\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 0)\\ &\quad + \Pr(\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 1)h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)},\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 1) \quad (70)\\ &= \Pr(\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 1)h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)},\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 1) \quad (71)\end{aligned}$$

since $h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)},\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 0) = h(O^{(i)}|Y_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)} = 0) = 0$. Substitute (69) into (71), we have:

$$\begin{aligned}& h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)})\\ &= \Pr(\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 1)\Pr(Y_{\boldsymbol{A}}^{(i)} = 1)h(O^{(i)}|Y_{\boldsymbol{A}}^{(i)} = 1,\boldsymbol{Y}_{\boldsymbol{a}}^{(i)} = 1)\\ &= \Pr(Y_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)} = 1)h(O^{(i)}|Y_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)} = 1)\\ &= h(O^{(i)}|Y_{\boldsymbol{A}\cup\{\boldsymbol{a}\}}^{(i)}).\end{aligned}$$

From Proposition 3, $-h(O^{(i)}|\boldsymbol{Y}_{\boldsymbol{A}}^{(i)})$, and hence $-h(O^{(i)}|Y_{\boldsymbol{A}}^{(i)})$, are monotone submodular. Consequently, it follows from [45, pp.272] that $V_2(\boldsymbol{A})$ is monotone submodular because it is a positive linear combination $-h(O^{(i)}|Y_{\boldsymbol{A}}^{(i)})$ □.