

# Measurement Simplification in $\rho$ -POMDP with Performance Guarantees

Tom Yotam  and Vadim Indelman 

**Abstract**—Decision making under uncertainty is at the heart of any autonomous system acting with imperfect information. The cost of solving the decision-making problem is exponential in the action and observation spaces, thus rendering it unfeasible for many online systems. This article introduces a novel approach to efficient decision making, by partitioning the high-dimensional observation space. Using the partitioned observation space, we formulate analytical bounds on the expected information-theoretic reward, for general belief distributions. These bounds are then used to plan efficiently while maintaining performance guarantees. We show that the bounds are adaptive and computationally efficient, and that they converge to the original solution. We extend the partitioning paradigm and present a hierarchy of partitioned spaces that allows greater efficiency in planning. We then propose a specific variant of these bounds for Gaussian beliefs and show a theoretical performance improvement of at least a factor of 4. Finally, we compare our novel method to other state-of-the-art algorithms in active simultaneous localization and mapping scenarios, in simulation and in real experiments. In both cases, we show a significant speedup in planning with performance guarantees.

**Index Terms**—Autonomous agents, localization, mapping, partially observable Markov decision process.

## I. INTRODUCTION

AUTONOMOUS agents must operate with imperfect information about their environments, dynamics, and measurements. *Belief space planning* (BSP) is one of the fundamental problems one must solve for these autonomous agents to interact with the environment successfully. The problem is modeled such that we maintain a probability density function over the true state of the agent, which is unknown. We then reason about the evolution of this distribution in the future for different actions and possible observations. BSP uses two main models to evolve the distribution over the state: motion and measurement models. The two models often have very distinct and different effects. The motion model introduces noise by state uncertainty. The measurement model on the other hand, although noisy by

Manuscript received 6 April 2024; accepted 12 June 2024. Date of publication 5 July 2024; date of current version 26 July 2024. This work was supported in part by the Israel Science Foundation, in part by the U.S. National Science Foundation/U.S.–Israel Binational Science Foundation, and in part by the Israeli Smart Transportation Research Center. This article was recommended for publication by Associate Editor R. Murrieta-Cid and Editor J. Civera upon evaluation of the reviewers’ comments. (*Corresponding author:* Vadim Indelman.)

Tom Yotam is with the Faculty of Mathematics, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: tomyo@campus.technion.ac.il).

Vadim Indelman is with the Department of Aerospace Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: vadim.indelman@technion.ac.il).

Digital Object Identifier 10.1109/TRO.2024.3424018

itself, usually provides valuable information about the agent’s pose and the map of the environment, which in turn reduces state uncertainty.

One advantage of using the BSP formulation is the capacity to incorporate belief-dependent reward functions, particularly, information-theoretic rewards, which is important for tasks such as search and rescue, informative path planning, and active classification. The ability to measure belief uncertainty and reduce it is key in solving such tasks; however, it comes with added computational complexity, especially when the observation space is high dimensional. In a typical active *simultaneous localization and mapping* (SLAM) setting, future observations may include hundreds or even thousands of landmarks. Moreover, in a visual-based partially observable Markov decision process (POMDP) setting, even a single observation can be high dimensional when the measurement model uses raw image inputs.

Solving the corresponding POMDP problem involves reasoning about different actions or policies and, for each, accounting for different possible observations. This leads to exponential growth of the posterior beliefs, which in turn makes the planning problem NP-hard [19].

A possible approach to addressing this issue is to simplify the planning problem. One specific simplification method of interest is forming analytical bounds on the expected reward and using the bounds to decide on the optimal action. Given that the bounds are easier to calculate than the expected reward, planning becomes more efficient.

In this work, we present a novel approach of simplification of the POMDP planning problem, specifically to the *multivariate observation space*. When the observation space is high dimensional, the calculation of expected information-theoretic rewards becomes expensive. We show that by partitioning the random variables that model future possible measurements into sets, this calculation becomes more efficient by developing novel bounds on the expected reward.

To illustrate, consider a factor graph representation for a given belief at planning time  $t = 2$ , as in Fig. 1. In this toy example, we sketch out how in planning, a specific action leads to three new random observations involving different landmarks. One can think of a simplified setting, which takes into account only a subset of these random observations. This subset of observations is a partition of the multivariate random variable representing the three original observations. This multivariate random variable defines in this case the observation space for this specific time step, such that its partitioning is a partition of the observation space. Similarly, in visual-based POMDP setting, a partition of the observation space may correspond to a partition of the random variables representing future image pixels into subsets.

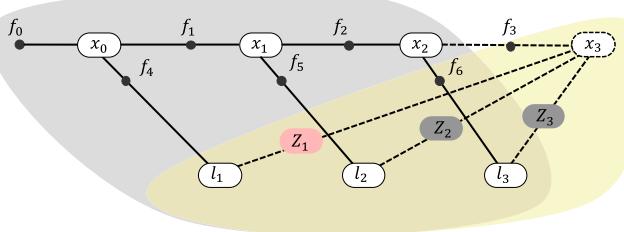


Fig. 1. Prior factor graph is shown in the gray blob; considering an action that leads to the factor  $f_3$ , we see the posterior factor graph in the yellow blob. The posterior graph includes the future + random measurements  $Z_1, Z_2$ , and  $Z_3$ . Different sets of random measurements are assigned different colors, which represents one possible partitioning. These sets are used to bound the conditional entropy of the entire posterior graph.

Specifically, our main contributions are as follows. We introduce the novel concept of observation space partitioning. Using the partitioned space, we show the relation to the original problem by deriving analytical bounds on the expected entropy that hold for all families of belief distributions. We present a partition tree that allows greater efficiency as we go down its hierarchy. We show that these bounds are adaptive and computationally cheaper, and that they converge to the original solution. Moreover, we show one possible realization of this general framework for an active SLAM scenario involving multivariate Gaussian distributions and present a hierarchy of efficient implementations. The speedup ranges from a factor of 4 for the least efficient one, and a speedup from a cube to linear time for the most efficient one. We demonstrate the use of this framework in both simulated and real experiments with a significant speedup.

## II. RELATED WORK

POMDP has been widely used as a model for decision making under uncertainty, despite the fact that obtaining the optimal solution for the planning problem is known to be intractable [9], [19]. While the standard POMDP is formulated with state-dependent reward functions, it is possible to extend this framework to include belief-dependent reward as well; examples for such frameworks are  $\rho$ -POMDP proposed in [1] and BSP in [20] and [29]. This extension is essential for tasks such as terrain monitoring [21], information gathering [26], and active SLAM [14]. Various approximation methods for solving POMDPs were proposed; however, even approximating a solution is challenging since most real-world problems incorporate continuous spaces. A sparse tree representation combined with a Monte Carlo tree search was explored in [25], in order to approximate near-optimal policies; however, it is not suitable for information-theoretic rewards. A particle-based approach to represent the belief was taken by Sunberg and Kochenderfer [27], proposing two different algorithms, one of them catering specifically to belief-dependent rewards. An abstraction of the observation model was studied in [2], for a low-dimensional state-space and nonparametric beliefs, allowing speedup compared with other approximation methods. Similarly to our method, it uses the observations as a means for simplification, where the main difference is that the mechanism for simplification is clustering of observation samples (post hoc), while our method is aimed at the observation space directly, by partitioning the high-dimensional observation space and reducing its dimensionality.

Early works have identified the importance of quantifying the information contained in observations. The quantitative value of possible measurements is presented in [5], in an effort to incorporate this value measure in inference. While it did allow for better selection of measurements using heuristics, it did not provide any optimality guarantees. Using the submodularity of mutual information, Krause et al. [17] formulated a near-optimal algorithm for sensor placement, which can alternatively cast as a myopic planning algorithm; however, it was limited to Gaussian processes.

Probabilistic graphical models have seen substantial traction in the world of inference; one of the most popular one was introduced by Kaess et al. [11] and was later extended in [10]. The latter utilized the structure of SLAM problems to be encoded in a Bayes tree, which allowed for an incremental update of the posterior belief with incoming information. The computational complexity for big SLAM problems is still very high, and there have been many works that have tried to reduce the computational complexity of inference, considering probabilistic graphical models. Some of the more popular SLAM methods using graphical models were introduced: Khosoussi et al. [13] presented a graph-theoretic approach to the problem of designing sparse reliable pose-graph SLAM in the context of measurement selection, Carlevaris-Bianco et al. [4] and Kretzschmar and Stachniss [18] showed methods for compressing a factor graph, and Zhang and Vela [30] reviewed how feature selection based on some defined scores can improve localization and data association, proposing a greedy algorithm that relies submodularity as well.

Other works have studied simplification methods for planning problems: Smith and Simmons [24] presented a heuristic-based bound on the value function, to guide local updates; a belief compression method was proposed in [22], but it lacked guarantees on planning performance. Several works have put forward simplified methods while providing guarantees: for a Gaussian high-dimensional state, Indelman [7] proposed a transformation of the original information space to a conservative one, by decoupling all state variables. More general approaches were studied in [6] and [31]; the former outlined a theoretical framework for simplification in general while demonstrating said framework for a sparse approximation of the initial belief, while the latter studied a simplification in risk averse planning, while considering a distributional perspective. A multilevel adaptive simplification framework considering belief-dependent rewards was developed in [28] and [32], based on novel bounds [28] on a differential entropy estimator [3] that utilize a reduced number of state samples. However, similarly to many nonparametric methods, these works aimed at a low-dimensional state-space setting. These works and other simplification methods are complimentary to ours and can be used alongside one another, as none of these works considered simplification to the observation space itself.

## III. NOTATIONS AND PRELIMINARIES

### A. $\rho$ -POMDP

A discrete-time POMDP models an agent decision process by outlining the dynamics of the interaction between the agent and its environment. It is defined as the tuple  $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, \rho)$ , consisting of state, action, and observation spaces, transition

and observation models, and a reward function. We assume a Markovian transition model, i.e.,  $T(X, a, X') = \mathbb{P}(X'|X, a)$ , and that each measurement is conditionally independent given the state, i.e.,  $O(X, z) = \mathbb{P}(z|X)$ .

Since the agent only observes the environment through noisy measurements, it must maintain a probability distribution over the true state; we denote this distribution as a *belief*. The belief and the propagated belief are defined as follows:

$$b_k \triangleq b[X_k] = \mathbb{P}(X_k | z_{0:k}, a_{0:k-1}) \triangleq \mathbb{P}(X_k | h_k) \quad (1)$$

$$b_k^- \triangleq b[X_k^-] = \mathbb{P}(X_k | z_{0:k-1}, a_{0:k-1}) \triangleq \mathbb{P}(X_k | h_k^-). \quad (2)$$

At each discrete-time step, this belief is updated with new motion and observation information according to Bayes' rule. Given an action  $a_k$  and observation  $z_{k+1}$ , the belief is updated according to

$$b_{k+1} = \eta \int_{X_k} \mathbb{P}(X_{k+1} | X_k, a_k) \mathbb{P}(z_{k+1} | X_{k+1}) b_k dX_k$$

where  $\eta$  is a normalization constant. A policy  $\pi : \mathcal{B} \mapsto \mathcal{A}$  maps belief states to actions. Usually, only state-dependent rewards are considered in the POMDP setting. BSP and later on  $\rho$ -POMDP extend the POMDP model to include belief-dependent rewards. For some finite planning horizon  $\ell$ , the *value* of a policy  $\pi$  is defined as the expected cumulative reward received by following  $\pi$  with initial belief  $b_k$ :

$$V^\pi(b_k) = \rho(b_k, \pi_k(b_k)) + \mathbb{E}_{Z_{k+1:k+\ell}} \left[ \sum_{i=k+1}^{k+\ell} \rho(b_i, \pi_i(b_i)) \right]. \quad (3)$$

Solving a POMDP is equivalent to finding the optimal policy  $\pi^*$  such that the value function is maximized.

A general form of the reward function  $\rho(b_i, \pi_i(b_i))$  can be expressed as a sum of a belief-dependent component  $R(b_i)$  and a state-dependent component  $R^X(b_i, \pi_i(b_i))$

$$\rho(b_i, \pi_k(b_i)) = R(b_i) + \alpha \cdot R^X(b_i, \pi(b_i)) \quad (4)$$

where  $\alpha$  is some weight. The belief-dependent component could correspond to an information-theoretic reward, such as differential entropy and information gain. The state-dependent component can be expressed as  $R^X(b_i, \pi(b_i)) = \mathbb{E}_{X_i \sim b_i} [r(X_i, \pi_i(b_i))]$ .

For instance,  $r(X_i, \pi_i(b_i))$  could represent distance to goal or obstacle and control effort.

In most cases,  $R(b_i)$  is the computationally expensive component, and as such, it is the focal point of this work. We consider a particular instance of this reward function, namely, differential entropy

$$R(b) \triangleq -\mathcal{H}(X) \equiv \mathbb{E}_{X \sim b} (\log b[X]) \quad (5)$$

where  $X$  is a random variable distributed according to  $b[X]$ .

If both  $X$  and  $Z$  are treated as random variables, the expected reward becomes the conditional entropy of these random variables, i.e.,

$$\mathbb{E}_Z[R(b)] = -\mathcal{H}(X | Z) = -\mathbb{E}_Z[\mathcal{H}(X | Z = z)]. \quad (6)$$

Thus, the expected reward at each  $i$ th look ahead step can be equivalently written as

$$\mathbb{E}_{Z_{k+1:i}} [R(b_i, \pi(b_i))] = -\mathcal{H}(X_i | Z_{k+1:i}) \quad (7)$$

where the future observations are drawn from the distribution  $\mathbb{P}(Z_{k+1:i} | b_k, \pi)$  and  $i \in [k+1, k+\ell]$ .

In this article, we consider an open-loop setting, as formulated in the next section.

### B. Active SLAM

Let  $x_k$  be the state of the agent at time  $k$ , and  $X_k$  be the joint state of the agent's trajectory and environment, e.g., landmarks, up to, and including time  $k$ . We define  $z_k \triangleq \{z_k^0, \dots, z_k^m\}$  as the set of all measurements observed at time  $k$ , and  $z_{0:k} \triangleq \{z_0, \dots, z_k\}$  as the set of all measurements until time  $k$ . Similarly, we define  $a_{0:k} \triangleq \{a_0, \dots, a_k\}$  as the set of all actions until time  $k$ . Assuming static landmarks, the motion of the agent, and the observations it receives are modeled as:

$$x_{k+1} = f(x_k, a_k) + w_k, \quad z_k = h(X_k) + v_k \quad (8)$$

where  $f$  and  $h$  are some deterministic functions, and  $w_k$  and  $v_k$  are their process noise, respectively.

We denote the data association vector at time  $k$  as  $\beta_k$ . The dimensionality of  $\beta_k$  is equal to  $X_k$  excluding  $x_k$  and is composed of binary entries, where each entry indicates whether the corresponding state was involved in a measurement at that given time step. We assume that each measurement involves the current pose  $x_i$ , such that  $\beta$  does not account for it. For example, at time step  $k = 3$ , for a prior state vector of dimensionality 5 and a measurement involving the third and fifth components of the state (e.g., observation of two landmarks),  $\beta_3 = (0 \ 0 \ 1 \ 0 \ 1)^T$ .

Alternatively, we can express the objective function using the data association vector. For a given action sequence  $a_{k:k+\ell}$ , the objective function is defined as

$$J(b_k, a_{k:k+\ell}) \triangleq \mathbb{E}_{\tilde{\beta}} \left[ \mathbb{E}_{\tilde{Z}|\tilde{\beta}} \left[ \sum_{i=k+1}^{k+\ell} R(b_i, a_i) \right] \right] \quad (9)$$

where  $\tilde{\beta} \triangleq \beta_{k+1:k+\ell}$  and  $\tilde{Z} \triangleq Z_{k+1:k+\ell}$ . In this scenario,  $\tilde{\beta}$  dictates the number of measurements and the states involved, while  $\tilde{Z}$  encodes the information about the distribution of those measurements. Combining Bayes rule and the properties of the models, we can factorize a posterior belief  $b[X_{k+\ell}]$  given  $\tilde{\beta}$  into prior belief, motion, and measurement factors

$$b[X_{k+\ell}] \propto b[X_k] \prod_{i=k+1}^{k+\ell} \mathbb{P}(x_i | x_{i-1}, a_{i-1}) \mathbb{P}(z_i | X_i, \beta_i) \quad (10)$$

where

$$\mathbb{P}(z_i | X_i, \beta_i) = \prod_{j=1}^{m_i(\beta_i)} \mathbb{P}(z_{i,j} | x_i, X_i^{\beta_i(j)}) \quad (11)$$

where  $m_i$  is the number of measurements at the  $i$ th time step, and  $X_i^{\beta_i(j)}$  represents the involved state in the  $j$ th measurement, at the  $i$ th time step, both as a function of  $\beta_i$ . We denote the set of all involved state variables for a given time step as

$$X_k^{\text{inv}} = \{X_k^{\beta_k(j)} | j \in \mathcal{J}\} \quad (12)$$

where  $\mathcal{J} = \{1, 2, \dots, m_k(\beta_k)\}$ .

When the models in (8) are linear, with zero-mean Gaussian noise, i.e.,  $w_k \sim \mathcal{N}(0, W_k)$  and  $v_k \sim \mathcal{N}(0, V_k)$ , and the prior belief is Gaussian, it can be shown that the posterior belief is also Gaussian. In such a case, the entropy of the posterior belief

can be expressed as

$$\mathcal{H}(X) = \frac{1}{2}(\ln |\Sigma| + N \ln(2\pi e)) \quad (13)$$

where  $b[X] = \mathcal{N}(\mu, \Sigma)$ , with mean  $\mu \in \mathbb{R}^N$  and covariance matrix  $\Sigma \in \mathbb{R}^{N \times N}$ . The above also holds when the belief is modeled or approximated as Gaussian for nonlinear models. The inverse of the covariance is known as the information matrix, such that  $\Sigma_k^{-1} = \Lambda_k$ . At each given time step, the information matrix of a posterior belief  $b_{k+1}$  can be decomposed to

$$\Lambda_{k+1} = \Lambda_k^{\text{Aug}} + F^T W_{k+1}^{-1} F + H^T V_{k+1}^{-1} H \quad (14)$$

where  $\Lambda_k^{\text{Aug}}$  is the prior information matrix of  $b_k$ , augmented with zeros to accommodate new states,  $W_{k+1}$  and  $V_{k+1}$  are the noise covariance matrices of the motion and measurement models, respectively, and  $F \triangleq \nabla f$  and  $H \triangleq \nabla h$  are the Jacobians of the motion and measurement functions, respectively. The stacked matrices of  $F^T W_{k+1}^{-1} F$  and  $H^T V_{k+1}^{-1} H$  are denoted as the collective Jacobian  $\tilde{A}_k$ ; see [8] for details. If we combine the collective Jacobians of consecutive time steps, i.e.,  $\tilde{A}_{k+1}, \tilde{A}_{k+2}, \dots, \tilde{A}_i$ , we get the following update rule:

$$\Lambda_i = \Lambda_k^{\text{Aug}} + \tilde{A}_{k+1:i}^T \cdot \tilde{A}_{k+1:i} \quad (15)$$

where  $\tilde{A}_{k+1:i}$  is the collective Jacobian of the motion and measurement factors of (10), from the time step  $k+1$  until  $i$ .

For the sake of readability, we drop the notation of the history  $h_i$  from now on, but assume that all distributions of a given time step are conditioned on the history available at the beginning of the planning session. We denote the collective Jacobian of a given horizon  $\tilde{A}_{k+1:i}$  simply as  $A_i$ .

#### IV. PROBLEM FORMULATION AND APPROACH

In the following section, we show how to use observation space partitioning to simplify the BSP problem. In order to choose the optimal action from a pool of candidate actions, one needs to evaluate the expected reward function  $\mathbb{E}_{Z_{k:i}}(\rho(b_i, a_i))$  at each future time instant  $i$  in the planning horizon  $\ell$  for each action sequence  $a_{k:k+\ell}$ . Instead, one can evaluate bounds on the expected reward function as a proxy

$$\mathcal{LB}_i^\rho \leq \mathbb{E}_{Z_{k:i}}(\rho(b_i, a_i)) \leq \mathcal{UB}_i^\rho. \quad (16)$$

In this work, we focus on bounding the information-theoretic reward  $R(b_i)$ , since it is typically the computational bottleneck with respect to the state-dependent reward

$$\mathcal{LB}_i \leq \mathbb{E}_{Z_{k:i}}(R(b_i)) \leq \mathcal{UB}_i \quad (17)$$

and thus

$$\mathcal{UB}_i^\rho \triangleq \mathcal{UB}_i + \alpha \mathbb{E}_{Z_{k:i}}[R^X(b_i, a_i)] \quad (18)$$

$$\mathcal{LB}_i^\rho \triangleq \mathcal{LB}_i + \alpha \mathbb{E}_{Z_{k:i}}[R^X(b_i, a_i)]. \quad (19)$$

Note that, considering an open-loop setting, the expected state-dependent reward  $\mathbb{E}_{Z_{k:i}}[R^X(b_i, a_i)]$  component in  $\rho(b_i, a_i)$  does not depend on future observations, which marginalize out: since  $R^X(b_i, a_i) = \mathbb{E}_{X_i \sim b_i}[r(X_i, a_i)]$ ,

$\mathbb{E}_{Z_{k:i}}[R^X(b_i, a_i)]$  can be expressed as

$$\int_{Z_{k:i}} \mathbb{P}(Z_{k:i} | b_k, a_{k:i}) R^X(b_i, a_i) dZ_{k:i}$$

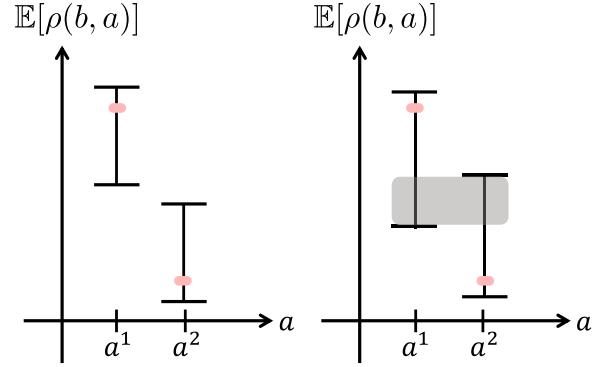


Fig. 2. Expected reward is shown in red within its bounds. On the left, we can select the optimal action based on the bounds alone; on the right, the worst-case loss is shaded in gray.

$$\begin{aligned} &= \int_{X_i} \mathbb{P}(X_i | b_k, a_{k:i}) r(X_i, a_i) dX_i \\ &= \mathbb{E}_{X_k \sim b_k} \mathbb{E}_{X_{k+1} \sim \mathbb{P}(\cdot | X_k, a_k)} \cdots \mathbb{E}_{X_i \sim \mathbb{P}(\cdot | X_{i-1}, a_{i-1})} [r(X_i, a_i)]. \end{aligned}$$

Therefore, the bounds from (18) assume the form

$$\mathcal{UB}_i^\rho \triangleq \mathcal{UB}_i + \alpha \mathbb{E}_{X_i \sim \mathbb{P}(\cdot | b_k, a_{k:i-1})} [r(X_i, a_i)] \quad (20)$$

$$\mathcal{LB}_i^\rho \triangleq \mathcal{LB}_i + \alpha \mathbb{E}_{X_i \sim \mathbb{P}(\cdot | b_k, a_{k:i-1})} [r(X_i, a_i)] \quad (21)$$

which means that the considered simplification of the observation space does not have an impact on the state-dependent reward component in an open-loop setting, which in any case does not depend on future observations.

As stated, this result is only valid for the open-loop setting considered herein, while in a closed-loop setting, we would remain with the bounds (18) and (19) (with appropriate replacement of actions to policies; see also [2]).

In the same manner, we can bound the objective function by summing up the bounds over the expected reward function for each of the time steps

$$\sum_{i=k+1}^{k+\ell} \mathcal{LB}_i^\rho \leq J(b_k, a_{k:k+\ell-1}) \leq \sum_{i=k+1}^{k+\ell} \mathcal{UB}_i^\rho. \quad (22)$$

Under the assumption that these bounds can be efficiently calculated, it is easier to select actions based on the reward bounds; see illustration in Fig. 2. We can think about two distinct actions  $a^1$  and  $a^2$ . For each action, we calculate the expected reward bounds and face two cases: in the first, the bounds do no overlap and we can select the optimal action; in the second, the bounds do overlap, and we choose between tightening the bounds such that they do not overlap and selecting an action while bounding the loss. Previous works have developed such bounds considering various simplification methods, as discussed in Section II.

In this work, we put forward a fundamental simplification that applies to the observation space itself. Specifically, we propose a partitioning of the observation space and develop expected reward bounds that are a function of this partitioning.

##### A. Partitioning of a Multivariate Observation Space

Consider a multivariate random variable  $Z$  that represents future observations and the corresponding observation space  $\mathcal{Z}$ .

In general,  $Z$  can be represented by

$$Z = (Z^1, Z^2, \dots, Z^m) \quad (23)$$

where  $Z^i$  is a random variable defined by a given measurement model, and  $m$  is the number of such random variables. We can now partition  $Z$  into different subset of components, for example, consider the partitioning  $Z^s \in \mathcal{Z}^s$  and  $Z^{\bar{s}} \in \mathcal{Z}^{\bar{s}}$ , such that

$$\begin{aligned} Z^s &= \{Z^1, Z^2, \dots, Z^n\} \\ Z^{\bar{s}} &= \{Z^{n+1}, Z^{n+2}, \dots, Z^m\} \end{aligned} \quad (24)$$

where  $Z = Z^s \cup Z^{\bar{s}}$ , and their corresponding subspaces  $\mathcal{Z} = \mathcal{Z}^s + \mathcal{Z}^{\bar{s}}$ .

In the subsequent section, we derive bounds on the expected reward that are a function of this partitioning, such that  $\mathcal{LB}$  and  $\mathcal{UB}$  from (17) become

$$\mathcal{LB}_i(b_i, Z_{k:i}^s, Z_{k:i}^{\bar{s}}) \quad (25)$$

$$\mathcal{UB}_i(b_i, Z_{k:i}^s) \quad (26)$$

where  $Z_{k:i}^s$  and  $Z_{k:i}^{\bar{s}}$  are subsets of measurements from time step  $k$  to  $i$ . The partitioning can be applied (but not limited) to two different observation spaces—a raw measurement such as pixels in an image, or to a SLAM scenario where the measurement model is defined by (8), and  $\beta$  dictates the dimension of the measurements, e.g., number of observed landmarks considering a future camera pose.

To illustrate the computational advantage of these bounds, we apply partitioning to a raw image measurement of size  $20 \times 20$  binary pixels. Each pixel is represented by a random variable  $Z^{x,y} \in \{0, 1\}$ , where  $x$  and  $y$  denote the pixel location on the sensor, and  $Z \in \mathcal{Z} \subseteq (\mathbb{F}_2)^{400}$ . We must consider all the different permutations for each of those pixels,  $2^{400}$  in total, which defines  $|\mathcal{Z}|$  in this case. For example, if we partition  $Z^s$  to represent the left half of the image and  $Z^{\bar{s}}$  to represent the right half, we need only to consider  $2^{200}$  permutations for  $Z^s$  and another  $2^{200}$  for  $Z^{\bar{s}}$  and  $2^{201}$  in total.

1) *Hierarchical partitioning:* There can be higher levels of partitioning, breaking down further a given measurement set into two sets. To encode this partitioning scheme, we index the partitioning depth and the number of nodes at a given depth. Each partitioned set is given a unique encoding denoted as  $Z^{n_i|m_j}$ , where  $n$  is the node number at the  $i$ th partitioning level, and  $m$  is the node number at the parent partitioning level  $j$ . We note this slight abuse of notation in regard to (24), but it should be clear from the context which notation is used. If two sets share a parent set, we consider them a base subset and its complement. For example,  $Z^{4_3|1_2}$  represents the fourth node of level 3, where the parent set is the first node of level 2. In this new notation,  $Z^s$  and  $Z^{\bar{s}}$  become  $Z^{1_1|1_0}$  and  $Z^{2_1|1_0}$  or equivalently  $Z^{1_1|1_0}$  and  $Z^{1_1|1_0}$ . Overall, for  $Z \in \mathbb{R}^m$ , it is possible to create a partition hierarchy of depth  $\log_2 m$ , as illustrated in Fig. 3.

## B. BOUNDS ON EXPECTED ENTROPY

In this section, we use measurement partitioning (MP) to derive novel information-theoretic reward bounds considering arbitrary distributions; we begin by introducing some helpful lemmas.

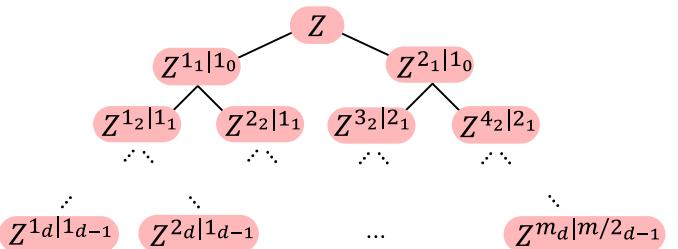


Fig. 3. Illustration of a possible partition tree. At each level of partitioning, we split a measurement set into two. For  $Z \in \mathbb{R}^m$ , the depth of the tree is  $d = \log_2 m$ .

*Lemma 1:* The conditional entropy can be factorized as

$$\mathcal{H}(X|Z) = \mathcal{H}(Z|X) + \mathcal{H}(X) - \mathcal{H}(Z). \quad (27)$$

*Proof:* For two random variables that have a joint entropy  $\mathcal{H}(X, Z)$ , we know that conditioning on  $Z$  yields  $\mathcal{H}(X, Z) = \mathcal{H}(X|Z) + \mathcal{H}(Z)$ . Similarly, conditioning on  $X$  yields  $\mathcal{H}(X, Z) = \mathcal{H}(Z|X) + \mathcal{H}(X)$ . Combining both equations and rearranging terms, we obtain the desired equality. ■

There are three quantities we need to evaluate in (27): 1) the entropy of the likelihood of a measurement, which is done via the measurement model; 2) the prior entropy; and 3) the entropy of the measurement given the history. The prior entropy is common to all actions and can be calculated once. The rest of the terms involve future random measurements, and we shall apply partitioning to these terms.

*Lemma 2:* Given two sets of expected measurements and the partitioning  $Z = Z^s \cup Z^{\bar{s}}$  from (24), the conditional entropy can be factorized as

$$\mathcal{H}(X|Z) = \mathcal{H}(Z^s|X) + \mathcal{H}(Z^{\bar{s}}|X) - \mathcal{H}(Z^s, Z^{\bar{s}}) + \mathcal{H}(X). \quad (28)$$

*Proof:* Using (27), we can rewrite the conditional entropy as  $\mathcal{H}(X|Z) = \mathcal{H}(Z^s, Z^{\bar{s}}|X) - \mathcal{H}(Z^s, Z^{\bar{s}}) + \mathcal{H}(X)$ . The measurements are independent given the state, such that  $\mathbb{P}(Z^s, Z^{\bar{s}}|X) = \mathbb{P}(Z^s|X)\mathbb{P}(Z^{\bar{s}}|X)$ . The entropy of two independent random variables is just the sum of individual entropies such that  $\mathcal{H}(Z^s, Z^{\bar{s}}|X) = \mathcal{H}(Z^s|X) + \mathcal{H}(Z^{\bar{s}}|X)$ . ■

Having established what MP looks like in (24), we can use it to bound the expected reward. The following two theorems present our main result.

*Theorem 1:* The conditional entropy can be bounded from above by

$$\mathcal{H}(X|Z) \leq \mathcal{UB} \triangleq \mathcal{H}(Z^s|X) + \mathcal{H}(X) - \mathcal{H}(Z^s). \quad (29)$$

*Proof:* It is not difficult to show that  $\mathcal{H}(X|Z^s) - \mathcal{H}(X|Z) = \mathcal{I}(X|Z^s; Z \setminus Z^s)$ . Recalling that the mutual information between two random variables is always nonnegative, we get

$$\mathcal{H}(X|Z) \leq \mathcal{H}(X|Z^s). \quad (30)$$

We denote this as the conditioning argument, i.e., conditioning on a random variable always reduces entropy, and refer to it later. Using Lemma 1, we get  $\mathcal{H}(X|Z^s) = \mathcal{H}(Z^s|X) + \mathcal{H}(X) - \mathcal{H}(Z^s)$ . ■

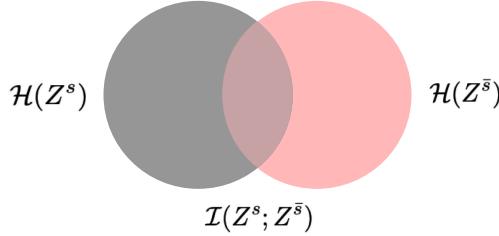


Fig. 4. Visualizing the marginal entropies of two measurement variables; the lower bound double counts the overlapping region, which is the mutual information between those variables.

*Theorem 2:* The conditional entropy can be bounded from below by

$$\begin{aligned} \mathcal{H}(X|Z) &\geq \mathcal{LB} \triangleq \mathcal{H}(Z^s|X) + \mathcal{H}(Z^{\bar{s}}|X) \\ &\quad - \mathcal{H}(Z^s) - \mathcal{H}(Z^{\bar{s}}) + \mathcal{H}(X). \end{aligned} \quad (31)$$

*Proof:* The joint entropy of the measurements can be written as  $\mathcal{H}(Z^s, Z^{\bar{s}}) = \mathcal{H}(Z^s|Z^{\bar{s}}) + \mathcal{H}(Z^{\bar{s}})$ . From the conditioning argument in (30), we get  $\mathcal{H}(Z^s|Z^{\bar{s}}) \leq \mathcal{H}(Z^s)$ , such that  $\mathcal{H}(Z^s, Z^{\bar{s}}) \leq \mathcal{H}(Z^{\bar{s}}) + \mathcal{H}(Z^s)$ . Rearranging the last inequality concludes the proof. ■

We can think of this lower bound from the perspective of mutual information: the difference between the original quantity  $\mathcal{H}(Z^s, Z^{\bar{s}})$  and the quantities  $\mathcal{H}(Z^s)$  and  $\mathcal{H}(Z^{\bar{s}})$  is exactly  $I(Z^s; Z^{\bar{s}})$ , such that the lower bound double counts the mutual information between the measurement sets (see Fig. 4).

In some specific cases, it might be easier to work with posterior beliefs, i.e., with expressions of the form  $\mathbb{P}(X|\cdot)$ . Such cases include situations where we have closed-form expression to a Bayesian belief update, for example when the belief is modeled as a Gaussian. In such cases, it makes sense to rearrange the bounds as follows.

*Corollary 1:* The conditional entropy can be bounded by

$$\mathcal{LB} = \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X) \quad (32)$$

$$\mathcal{UB} = \mathcal{H}(X|Z^s). \quad (33)$$

This can be obtained directly from (29) and (31) using Lemma 1.

We note that if one does not have access to the underlying belief distribution, one must resort to approximation of the expected reward, i.e., conditional entropy. This is usually done by sampling the closed-form expressions for the motion and measurement models. Such cases are outside of the scope of this work.

### C. BOUNDS WITH HIERARCHICAL PARTITIONING

In the previous section, we formed bounds based on a partitioning of measurement sets, by double counting the mutual information between said sets. This intuition applies to the hierarchical partitioning from Section IV-A as well, allowing us to create a hierarchical notion of the bounds, starting with the lower bound. To formulate this idea, we define a new operator

$$g(Z^s, Z^{\bar{s}}) = \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X) \quad (34)$$

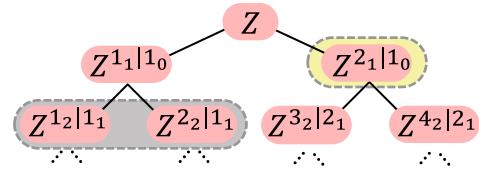


Fig. 5. Any combination of children nodes, that their union equals the parent node, can make up a lower bound on that parent node. Any child node at any depth can make up an upper bound on a parent node. For instance, the node highlighted in yellow can form an upper bound, while the nodes highlighted in gray and yellow can form a lower bound.

where  $g(Z, \emptyset) = g(\emptyset, Z) \triangleq \mathcal{H}(X|Z) - \mathcal{H}(X)$ , and  $g(\emptyset, \emptyset) \triangleq -\mathcal{H}(X)$ .

Using this operator, (32) can be expressed as  $\mathcal{LB} = g(Z^s, Z^{\bar{s}})$  and (33) can be expressed as  $\mathcal{UB} = g(Z^s, \emptyset) + \mathcal{H}(X)$ . Given that measurements sets can be hierarchically partitioned further, we can formulate their bounds.

*Theorem 3:* For the sets  $Z^s$  and  $Z^{\bar{s}}$ , and their respective children  $Z^{s_1}, Z^{s_2}, Z^{\bar{s}_1}$ , and  $Z^{\bar{s}_2}$ , the following holds:

$$\begin{aligned} g(Z^s, Z^{\bar{s}}) &\geq \\ g(Z^{s_1}, Z^{s_2}) + g(\emptyset, Z^{\bar{s}}) &\geq \\ g(Z^{s_1}, Z^{s_2}) + g(Z^{\bar{s}_1}, Z^{\bar{s}_2}) + g(\emptyset, \emptyset). \end{aligned}$$

*Proof:* From (32), we know that

$$\mathcal{H}(X|Z) \geq \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X).$$

Substituting  $\mathcal{H}(X|Z^s)$  instead of  $\mathcal{H}(X|Z)$  yields

$$\mathcal{H}(X|Z^s) \geq \mathcal{H}(X|Z^{s_1}) + \mathcal{H}(X|Z^{s_2}) - \mathcal{H}(X)$$

which proves the first inequality. Doing the same for  $\mathcal{H}(X|Z^{\bar{s}})$  proves the second inequality. ■

Each of the quantities in Theorem 3 is a lower bound on the expected reward by itself. As for the upper bound, the entropy of each of the child sets of a given set is an upper bound on the entropy of that given parent set.

*Theorem 4:* For the set  $Z^s$ , and its children  $Z^{s_1}$  and  $Z^{s_2}$ , the following holds:

$$\mathcal{H}(X|Z^s) \leq \mathcal{H}(X|Z^{s_1}) \wedge \mathcal{H}(X|Z^s) \leq \mathcal{H}(X|Z^{s_2}). \quad (35)$$

*Proof:* This follows directly from the conditioning argument. ■

We can use the above theorems to perform further partitioning of the measurement sets. Theorem 3 shows that we can mix different partition depths for the lower bound, while Theorem 4 shows that any node in the partition tree can be used to form upper bound; see partition tree Fig. 5 for example.

We also note that we have the ability to adaptively change those bounds, by moving between partitioning levels, as well as by moving measurements from one set to its compliment. In the next section, we show the convergence of the bounds, so by adaptively changing the bounds, we also control how tight they are.

### D. ANALYSIS OF THE BOUNDS

In this section, we analyze the properties of the derived bounds. We look at the convergence and monotonicity of the bounds, both as a function of the sets, for a given partition depth,

and as a function of the partition depth. We then examine the computational complexity of obtaining the bounds.

### 1) Convergence

For a given partition depth of  $d \in [1, \log_2 m]$ , we show that the bounds converge to the bounds of the parent depth,  $d - 1$ . In particular, when the partition depth is 1, the bounds converge to the original expected reward. For  $Z^s \cup Z^{\bar{s}} \subseteq Z$ , the upper bound converges when we add variables to the set  $Z^s$ , while the lower bound converges when we remove variables from the set  $Z^{\bar{s}}$  and add them to the set  $Z^s$ . We show the proof for the first partition depth, but it is valid for any arbitrary depth.

*Theorem 5:* If  $Z^s \rightarrow Z$  and  $Z^{\bar{s}} \rightarrow \emptyset$ , then  $\mathcal{H}(X|Z^s) \rightarrow \mathcal{H}(X|Z)$  and  $g(Z^s, Z^{\bar{s}}) \rightarrow \mathcal{H}(X|Z)$ .

*Proof:* Without loss of generality, assuming  $Z^s \in \mathbb{R}^n$  and  $Z^{\bar{s}} \in \mathbb{R}^{m-n}$  as in (24), we start with the upper bound,  $\mathcal{UB} = \mathcal{H}(X|Z^s)$ . Using the conditioning argument,  $\mathcal{H}(X|Z^s \cup \{Z^{n+1}\}) \leq \mathcal{H}(X|Z^s)$ , where  $Z^{n+1} \subseteq Z^{\bar{s}}$  but  $Z^{n+1} \not\subseteq Z^s$ . We continue adding variables to the set  $Z^s$  in this way, until  $Z^s \cup \{Z^{n+1}\} \cup \dots \cup \{Z^m\} = Z$  and  $\mathcal{H}(X|Z^s \cup \{Z^{n+1}\} \cup \dots \cup \{Z^m\}) = \mathcal{H}(X|Z)$ .

As for the lower bound,  $\mathcal{LB} = g(Z^s, Z^{\bar{s}}) = \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X)$ . We use the same argument, while working in the opposite direction with  $Z^{\bar{s}}$ :  $\mathcal{H}(X|Z^{\bar{s}}) \leq \mathcal{H}(X|Z^{\bar{s}} \setminus \{Z^{n+1}\})$ , where  $Z^{n+1} \subseteq Z^{\bar{s}}$ . We continue removing variables from the set  $Z^{\bar{s}}$  in this way until  $Z^{\bar{s}} \setminus \{Z^{n+1}\} \setminus \dots \setminus \{Z^m\} = \emptyset$  and  $\mathcal{H}(X|\emptyset) = \mathcal{H}(X)$ . Plugging into the expression for  $\mathcal{LB}$ , we get  $\mathcal{H}(X|Z) + \mathcal{H}(X|\emptyset) - \mathcal{H}(X) = \mathcal{H}(X|Z)$ . ■

### 2) Monotonicity

From the conditioning argument, we can see that  $\mathcal{UB}$  monotonically converges to the expected reward, or in different words, the entropy is monotone in the size of the measurement set. This holds true both across different partitioning depths and within a given partitioning depth. However, we cannot say the same about  $\mathcal{LB}$  since it depends on the change of both  $\mathcal{H}(X|Z^s)$  and  $\mathcal{H}(X|Z^{\bar{s}})$ . We always move measurements from one set to the other, such that these two quantities change in opposition to one another. We cannot say a priori what change is greater and thus cannot deduce monotonicity. In fact, it can be proven that this bound is nonmonotone using the submodularity of the entropy in the measurements, but it is beyond the scope of this work.

### 3) Computational Complexity

It is only rational to use the bounds instead of the full calculation of the expected reward function, if computing the bounds is cheaper computationally. We are now going to compare these two calculations.

For general distributions, we can compare the calculation as a function of the observation space size. The baseline calculation involves evaluating the following quantities (omitting history and actions, see Lemma 2):

$$\mathcal{H}(Z^s|X), \mathcal{H}(Z^{\bar{s}}|X), \mathcal{H}(X), \mathcal{H}(Z^s, Z^{\bar{s}}). \quad (36)$$

Using the bounds, we need to evaluate (see Theorems 1 and 2)

$$\mathcal{H}(Z^s|X), \mathcal{H}(Z^{\bar{s}}|X), \mathcal{H}(Z^{\bar{s}}), \mathcal{H}(Z^s), \mathcal{H}(X). \quad (37)$$

Since the prior entropy is not a function of the action or expected measurements, we can calculate it once for each planning session. Overall, the difference between the expected reward and the bounds boils down to the difference between the joint versus the marginal entropy of the measurements, i.e.,  $\mathcal{H}(Z^s, Z^{\bar{s}})$  versus  $\mathcal{H}(Z^{\bar{s}})$  and  $\mathcal{H}(Z^s)$ . The baseline calculation is

$$\mathcal{H}(Z) = - \int_{Z^s} \int_{Z^{\bar{s}}} \mathbb{P}(Z^s, Z^{\bar{s}}) \log \mathbb{P}(Z^s, Z^{\bar{s}}) dZ^s dZ^{\bar{s}}. \quad (38)$$

Assuming that the observation space is finite and countable, the cost of evaluating the integral terms is a function of the random variables. Evaluating (38) is of the order of  $O(|Z^s||Z^{\bar{s}}|)$ , while evaluating the simplified terms  $\mathcal{H}(Z^s)$  and  $\mathcal{H}(Z^{\bar{s}})$  is of the order of  $O(|Z^s| + |Z^{\bar{s}}|)$ .  $Z^s$  and  $Z^{\bar{s}}$  are entirely defined by the measurement model; see example in Section IV-A.

Note that, in practice, we do not have access to the joint or marginal distribution over the measurements, only the measurement model. Marginalization over the state yields

$$\begin{aligned} \mathcal{H}(Z) &= - \int_Z \int_X \mathbb{P}(Z, X) \log \int_{X'} \mathbb{P}(Z, X') dX' dZ dX \\ &= - \int_Z \int_X \mathbb{P}(Z|X) \mathbb{P}(X) \log \int_{X'} \mathbb{P}(Z|X') \mathbb{P}(X') dX' dZ dX. \end{aligned} \quad (39)$$

The simplification is of the same factor; only now it is magnified by the state vector size to the second power:  $O(|Z^s||Z^{\bar{s}}||\mathcal{X}^2|)$  for the baseline and  $O((|Z^s| + |Z^{\bar{s}}|)|\mathcal{X}^2|)$  for the bounds.

## E. HIGH-DIMENSIONAL STATE

In this section, we discuss what the measurement simplification looks like when the state space is high dimensional. Specifically, we consider an active SLAM formulation as in Section III-B. Utilizing data association information from (10) and (12), we can further simplify the bounds (29) and (31)

$$\begin{aligned} \mathcal{LB} &\triangleq \mathcal{H}(Z^s|X^{\text{inv}_s}) + \mathcal{H}(Z^{\bar{s}}|X^{\text{inv}_{\bar{s}}}) \\ &\quad - \mathcal{H}(Z^s) - \mathcal{H}(Z^{\bar{s}}) + \mathcal{H}(X) \end{aligned} \quad (40)$$

$$\mathcal{UB} \triangleq \mathcal{H}(Z^s|X^{\text{inv}_s}) + \mathcal{H}(X) - \mathcal{H}(Z^s) \quad (41)$$

where  $X^{\text{inv}_s}$  and  $X^{\text{inv}_{\bar{s}}}$  are the states involved in the random measurements  $Z^s$  and  $Z^{\bar{s}}$ , respectively, as determined by  $\beta$ .

## F. GAUSSIAN BELIEF

The bounds shown in the previous section hold for general belief distributions. In this section, we develop a specific form of the bounds, considering Gaussian distributions and a high-dimensional state in the context of active SLAM.

In the case of Gaussian distributions, we can replace the entropy term in (6) for its closed-form expression, combining (13) and (15). For a lookahead step  $i \in [k+1, k+\ell]$ , the expected entropy can be expressed as

$$\mathbb{E}_{Z_{k+1:i}} [\mathcal{H}(X_i|h_i)] = C - \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} \left[ \ln \left| \Lambda_k^{\text{Aug}} + A_i(z)^T A_i(z) \right| \right] \quad (42)$$

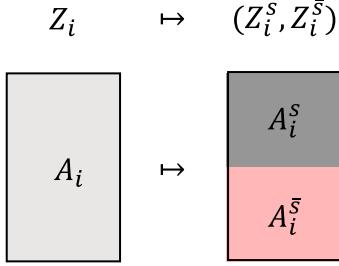


Fig. 6. Illustration of a possible partitioning of a measurement collective Jacobian, given some partitioning of  $Z_i$ .

where  $A_i(z) = \tilde{A}_i(z_{k+1:i})$  is the collective Jacobian for the entire planning horizon as a function of the measurements, and  $C \triangleq N \ln(2\pi e)$ .

For convenience, we define  $f(\Lambda, A) \triangleq |\Lambda + A^T A|$ . Since we are interested in the Jacobian of the measurement but not the motion factors, we choose to present the expected entropy in the following form:

$$\mathbb{E}_{Z_{k+1:i}} [\mathcal{H}(X_i | h_i)] = C - \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} [\ln f(\Lambda_k^{\text{AUG}-}, A_i(z))] \quad (43)$$

where  $\Lambda_k^{\text{AUG}-}$  is the propagated belief, i.e., without taking into account measurements until time step  $i$ , augmented with zeros, and the measurement vector defined in (23) becomes  $Z_{k+1:i}$ .

Utilizing (32), (33), and (42), we are ready to present the expected reward bounds for the Gaussian case. The bounds apply in expectation and are more efficient to calculate than state-of-the-art (SOTA) methods per sample of measurements under the expectation operator.

*Theorem 6:* For the  $i$ th look ahead step, the expected entropy of a Gaussian belief can be bounded by

$$\mathcal{LB} = C - \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} [\ln \frac{f(\Lambda_k^{\text{AUG}-}, A_i^s) \cdot f(\Lambda_k^{\text{AUG}-}, A_i^{\bar{s}})}{|\Lambda_k^{\text{AUG}-}|}] \quad (44)$$

$$\mathcal{UB} = C - \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} [\ln f(\Lambda_k^{\text{AUG}-}, A_i^s)] \quad (45)$$

where  $A_i^s$  and  $A_i^{\bar{s}}$  are the Jacobian rows associated with the measurements  $Z_i^s$  and  $Z_i^{\bar{s}}$ , respectively (see Fig. 6).

*Proof:* We have obtained a closed-form expression (42) for the expected entropy. If we plug in these expressions into (32), we get

$$\begin{aligned} \mathcal{H}(X_i | Z_i) &\geq C - \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} [\ln f(\Lambda_k^{\text{AUG}-}, A_i^s)] \\ &\quad + C - \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} [\ln f(\Lambda_k^{\text{AUG}-}, A_i^{\bar{s}})] \\ &\quad - C + \frac{1}{2} \mathbb{E}_{Z_{k+1:i}} [\ln |\Lambda_k^{\text{AUG}-}|]. \end{aligned}$$

In a similar way, using (33), we obtain the upper bound. ■

### 1) Estimation of Expected Entropy

An empirical expectation is generally calculated by sampling the observation model; for each such sampled observation, a posterior distribution of the belief needs to be obtained. The latter usually involves solving a nonlinear optimization problem via

some iterative linearization method. For each posterior belief, one can calculate the corresponding reward bounds using (44) and (45), where the expectation operator is approximated by samples. In this case, the performance guarantees are asymptotic, since one assumption that was used to obtain the bounds is only asymptotically valid, namely, the nonnegativity of the mutual information between measurements. We note that it is possible to formulate nonasymptotic guarantees in this case, but it is outside of the scope of this work and we leave it for future work.

A different approach would be to make a common assumption for the optimization process within planning, by taking a single iteration step. In the context of planning, this is usually a good approximation for the linearization point when the state prior is informative and is accurate when the measurement model uses a linear function or for maximum likelihood estimation for the measurements [8], [20]. Under the said assumption, the Jacobian is only a function of  $\beta$  and not of any particular measurement realization. When the Jacobian is independent of the actual measurement values, we can drop the expectation operator from (42) and still represent the expected entropy

$$\mathcal{H}(X_i | Z_{k+1:i}) = C - \frac{1}{2} (\ln |\Lambda_k^{\text{Aug}} + (A_i)^T \cdot A_i|). \quad (46)$$

In this case, the bounds in (44) and (45), as well as other SOTA calculations, should be computed only once per action, i.e., without the expectation operator.

### 2) Methods for Determinant Calculation

The computational cost of the bounds in the Gaussian case is the cost of evaluating the appropriate determinants  $|\Lambda_k^{\text{AUG}-} + (A_i^s)^T A_i^s|$  and  $|\Lambda_k^{\text{AUG}-} + (A_i^{\bar{s}})^T A_i^{\bar{s}}|$ . At a first glance, it is not clear that evaluating the bounds is indeed more efficient than evaluating the expected reward; we will now show one method, which is, in fact, efficient.

There are three main methods for calculating the entropy of the posterior belief: calculating the determinant of the posterior information matrix (denoted to as baseline), calculating the determinant of the posterior information matrix in its square root form (denoted as  $R$ ), and using the augmented matrix determinant lemma (rAMDL) [15], [16]. In essence, the rAMDL utilizes the well-known matrix determinant lemma, combined with some clever calculation reuse, to efficiently evaluate the determinant of the posterior information matrix. It requires a one-time calculation of some specific covariance entries, and its general form is as follows:

$$|\Lambda + A^T A| = |\Lambda| \cdot |\Delta| \cdot |(A_{\text{new}})^T \cdot \Delta^{-1} \cdot A_{\text{new}}| \quad (47)$$

where  $\Delta = I_m + A_{\text{old}} \cdot \Sigma \cdot (A_{\text{old}})^T$ ,  $\Sigma$  is the prior covariance matrix, and  $A_{\text{old}}$  and  $A_{\text{new}}$  are the blocks of the Jacobian matrix  $A$ , with respect to states at planning time (old) and states added by future actions (new), i.e.,  $A = [A_{\text{old}}, A_{\text{new}}]$ .

Applying MP to the baseline method is not efficient: the baseline cost is  $O(n^3)$ , where  $n$  is the dimension of  $\Lambda$ . Since  $n$  remains the same given a partition, the overall calculation would be worse than  $O(n^3)$ .

We choose to apply MP to the rAMDL and not  $R$  for two reasons: First, it has been shown that rAMDL is faster than  $R$  within planning [15], [16]. Second, applying MP in  $R$  does

not make sense—although calculating the determinants in  $R$  is  $O(n)$ , most of the computational burden lies in the update of the posterior  $R$  matrix. This update involves a QR decomposition, and by partitioning, we split it into two separate QR decompositions that, on average, are less efficient than the original one. We are then left with the most efficient method, which is rAMD.

### 3) Computational Complexity

One main insight is that the cost of calculating the determinant using rAMD (not including the one-time calculation which is common across all actions) is  $O(m^3)$ , where  $m$  is the dimension of  $A$ . In a typical high-dimensional scenario,  $m \ll n$ , and by partitioning the measurements, we reduce the dimension of  $A$  even further, so it becomes even more efficient.

Applying (47) to the determinants required for (44) and (45) results in

$$|\Lambda_k^{\text{AUG-}} + (A_i^s)^T A_i^s| = |\Lambda_k| \cdot |\Delta_k^s| \cdot |(A_{\text{new}}^s)^T (\Delta_k^s)^{-1} A_{\text{new}}^s|$$

where  $\Delta_k^s = I_m + A_{\text{old}}^s \cdot \Sigma_k \cdot (A_{\text{old}}^s)^T$ , and  $A_{\text{old}}^s$  and  $A_{\text{new}}^s$  are the old and new blocks of the matrix  $A_s$ , i.e.,  $A_s = [A_{\text{old}}^s, A_{\text{new}}^s]$ . In a similar way, we apply the rAMD to  $|\Lambda_k^{\text{AUG-}} + (A_i^s)^T A_i^s|$ .

The most efficient partitioning, for the first depth, is when  $s = \bar{s}$ , so the resulting matrices are of rank  $m/2$ . The cost of evaluating each of the determinants for the partitioned Jacobians is  $O(\frac{m^3}{8})$ . Compared to the rAMD, the cost is reduced from  $O(m^3)$  to  $O(\frac{m^3}{4})$  by using the bounds.

## V. RESULTS

In this section, we start off by demonstrating the properties of the bounds, as presented in Section IV-D, for a typical active SLAM scenario. We then show that compared to the other SOTA method, our approach is faster while achieving similar reward, both in a simulated scenario and in the real-world experiment.

### A. Setup

The setting considered is a planar high-dimensional SLAM, where the state includes past poses, current poses, and landmarks. The belief over the state is normally distributed, and the reward function is differential entropy. The action space is continuous, and each pose and landmark has three and two degrees of freedom, respectively. Thus, for a planning session at time instant  $k$  and planning horizon  $\ell$ , the joint state has the dimensionality of  $3(k + \ell) + 2L$ , where  $L$  is the number of mapped landmarks. As mentioned in Section IV-F1, in our simulations, we generate maximum-likelihood observations, although this is not a limitation of our approach. In all scenarios, the agent uses its sensor to map an unknown environment. In the simulation, we use a range and bearing sensor; the real experiment uses cameras and runs visual SLAM. After an initial mapping procedure, a prior belief containing poses and landmarks is formed. The agent then starts a planning session, where the objective is to reach a certain goal while minimizing the uncertainty over the high-dimensional state at the last planning step. A typical prior belief is illustrated in Fig. 7.

GTSAM 4.1.0 and Python 3.9.7 were used for the simulated scenarios, running on Ubuntu 18.04 and AMD Ryzen 7 3700X eight-core processor. For the real-world experiment, GTSAM

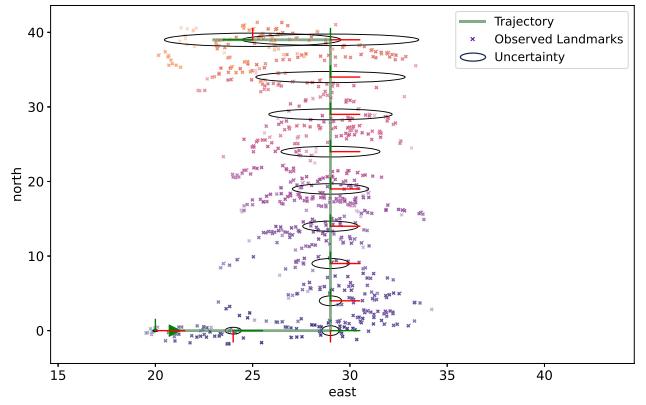


Fig. 7. Illustration of the prior belief showing a subset of the state landmarks. The uncertainty associated with the joint covariance matrix is shown for every fifth pose. The time of each observed landmark is encoded in color—from a dark one at the beginning to light at the end.

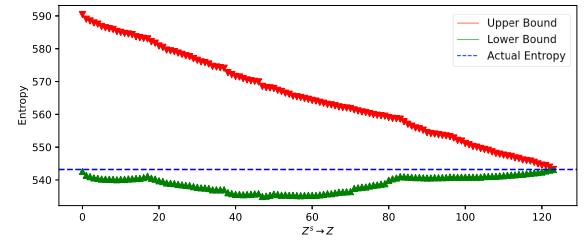


Fig. 8. Empirical demonstration of the bounds’ behavior, for  $Z^s \rightarrow Z$ .

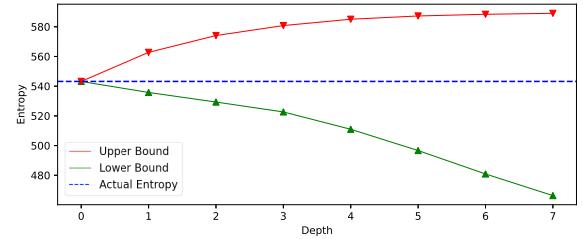


Fig. 9. Empirical demonstration of the bounds’ behavior, as  $d \rightarrow \log_2 m$ .

4.1.0 and Python 3.8 were used, running on Ubuntu 20.04 and Intel(R) Xeon(R) CPU E5-1620.

### B. Bound Analysis

1) *Inter Depth Properties:* Fig. 8 shows the properties of each of the bounds, as individual measurements are moved from one partition to the other, specifically, as  $Z^s \rightarrow Z$  and  $Z^{\bar{s}} \rightarrow \emptyset$ . We can see that both bounds converge to the actual expected entropy, the upper bounds does so monotonically, while the lower bounds does not, as expected. As proposed in Section IV-D, the lower bound does not converge monotonically because it is a function of the mutual information between the measurement partitions. Since the assignment to partitions is random, so is the value of the mutual information between them.

2) *Intra Depth Properties:* We show the properties of the bounds, as we go deeper in the partition tree. Fig. 9 shows, as expected, that the lower we go down the partition tree, the

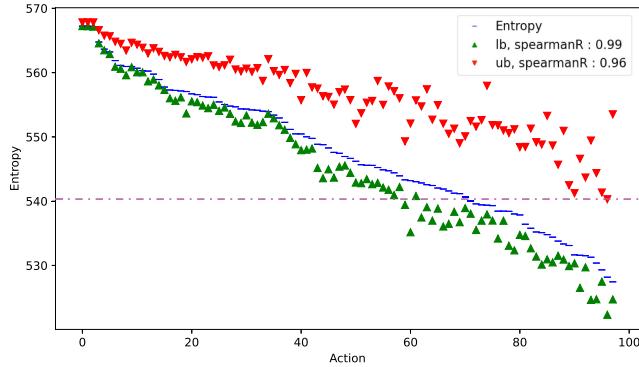


Fig. 10. Expected entropy bounds for each sampled action. All actions whose lower bound is above the purple dashed line are suboptimal and can be pruned safely.

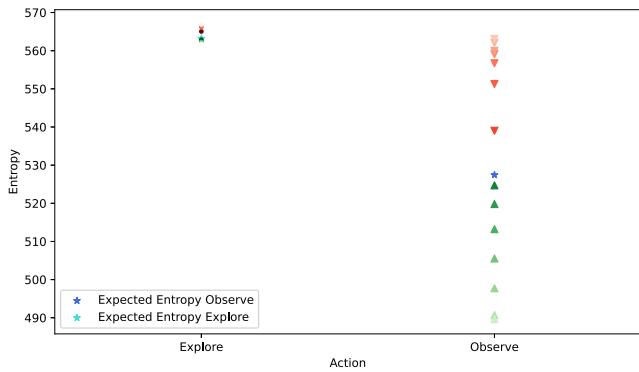


Fig. 11. Hierarchical bounds for two distinct actions “Explore” and “Observe.” The lighter the color, the more efficient the bounds are.

efficiency of the bounds comes at the expanse of their tightness. We can see that both bounds monotonically converge to the real expected reward as we go up the partition tree.

*3) Correlation and Tightness:* Here, we demonstrate the usefulness of the bounds in the general case, by presenting the expected reward bounds, for a set of myopic random actions. Looking at Fig. 10, we can see that although some bounds overlap, we can safely prune around 60% of the actions, which are suboptimal.

We can observe that the lower bound is very strongly correlated with the actual reward, while the upper bound shows slightly weaker correlation with the reward. Again, a possible explanation for this behavior is the random assignment of the individual observations to their corresponding set. Since the lower bound accounts for both sets, it is only a function of their mutual information, while the upper bound is a function of this random assignment.

Next, we compare two specific actions, taken from the previous random action set, by showing their respective hierarchical bounds. Fig. 11 shows that for actions that lead to rewards that are distinct, it is possible to select the optimal action based on a very efficient calculation of the bounds. Specifically, we compare an action that explores a new part of the map to another that reobserves previously seen landmarks.

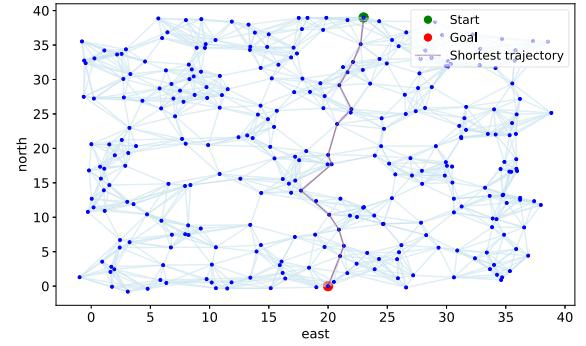


Fig. 12. PRM generator is used to construct a viable path from the agent’s pose at the start of the planning session to the desired goal. We then select a number of the shortest ones.

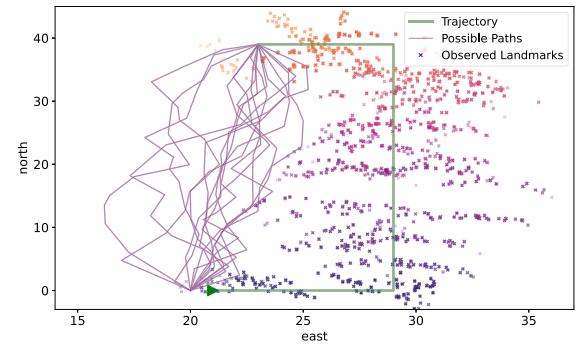


Fig. 13. Illustration of the simulated scenario, showing a subset of the landmarks and paths.

### C. Planning Using Bounds

*1) Simulation:* We now demonstrate the use of the bounds in a simulated SLAM scenario using the GTSAM library and a *probabilistic roadmap* (PRM) generator [12]; the goal is to demonstrate a typical use case in active SLAM setting. We start the planning session with a prior belief as in Fig. 7. We then use the PRM to randomly generate a set of candidate paths, shown in Fig. 12. As seen in Fig. 13, the paths start at the green dot and the goal shown in red, each path consisting of around 12 actions. The PRM samples are drawn from a uniform continuous distribution. The actions of a given path follow the trajectory dictated by the PRM. For each path from a set of the shortest ones, we evaluate the expected reward at the goal, and its bounds then choose the optimal one. We emphasize that our approach is applicable also to other methods for generating candidate action sequences and not limited to the above-described specific method.

We start by comparing our proposed method using MP to the rAMD. Since our method uses the rAMD to compute the determinants required for the bounds, many of the calculations are common and are not accounted for in the comparison. When compared to iSAM2, these will be accounted for. We compared the performance for three different scenarios, while two of them utilize replanning. In a replanning scenario, the agent chooses an optimal path, takes the first action, and redraws paths to the goal from the new pose. The number of replanning steps is 5; since the goal is fixed, each consecutive trajectory is shorter than its predecessor; see Fig. 14 for details.

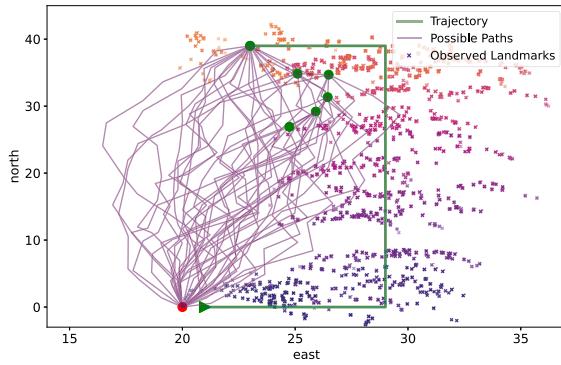


Fig. 14. Replanning scenario; each green dot represents the new pose for planning, and the goal is represented by a red dot.

TABLE I

TOTAL PLANNING TIME IN SECONDS (LOWER IS BETTER), NOT INCLUDING ALL COMMON ONE-TIME CALCULATIONS

# Paths	# Factors	RP	rAMDЛ	MP (ours)
100	2956	No	$11.521 \pm 0.537$	$6.888 \pm 0.155$
100	2956	Yes	$24.636 \pm 1.381$	$11.758 \pm 0.372$
100	5904	Yes	$84.376 \pm 14.458$	$32.069 \pm 4.913$

For all scenarios the same best action was calculated. Therefore, the objective function value, which corresponds in our case to the expected entropy at the end of the planning horizon, was identical for all methods.

From Table I, we can see that the speedup increases with the number of factors. That is expected since the speedup is correlated with the number of Jacobian rows. Since most of the prior factors relate to landmarks, a higher number of factors results in a higher number of observations. We can also see that the speedup is approaching the theoretical value of  $O(\frac{m}{4})$ , shown in Section IV-F3. This is because the higher the number of observations, the smaller the impact of the one-time overhead on the total planning time. Similar to all of the following scenarios, we use the bound to prune suboptimal paths; when the bounds overlap, we choose the path that has the lowest lower bound and bound the possible loss we might incur. For example, for the third scenario in Table I, the bound on the expected loss was 109.578, and the optimal expected reward was 5215.999. The ratio of loss divided by the optimal reward was 0.021, such that in the worst case, the trajectory chosen would set us off from the optimal one by 2%. Although unknowable in planning, in this specific case, as in many others, the lowest lower bound indeed corresponded the optimal expected reward, and the loss in practice was equal to zero.

Next, we compare our method to iSAM2 and rAMDЛ, including all one-time calculations, in two different scenarios. The first scenario uses a similar prior belief to the previous ones, while the second uses a different prior that includes obstacles. We use iSAM2 [10] to incrementally update the square root of the information matrix and then use the updated  $R$  matrix to calculate the reward as in Section IV-F2. Fig. 15 shows the bounds for the first simulated scenario, 2500 paths were evaluated without replanning, where we can see that about 85% of the suboptimal paths can be safely pruned using the bounds. Table II shows the total planning time for each method in the first scenario. We show separately the time it takes to recover the required entries from the prior covariance matrix in Table III: *worst case* is the time it takes to recover only the entries required for the states involved in the measurements for the set of all trajectories. Generally, the earlier in the robot's past trajectory the involved states are, the more of the covariance matrix entries we would need to recover, which is time consuming. However, in a scenario such as that, the time it would take to update the posterior  $R$  matrix needed for iSAM2 would be significant as well. Including the covariance recover, we can observe that our method is about 45% faster than rAMDЛ and about 65% faster than iSAM2.

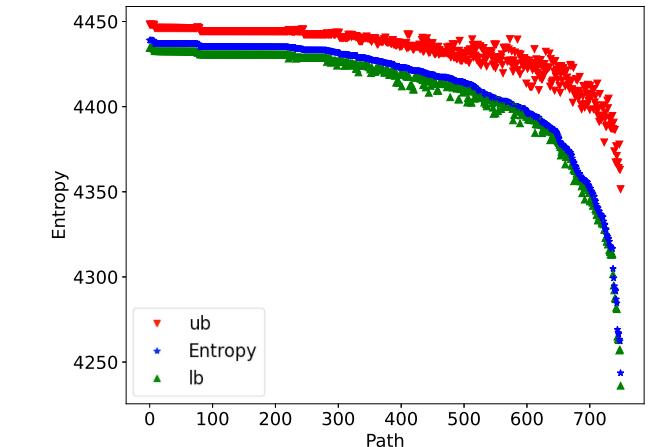


Fig. 15. Bounds for the first simulated scenario, showing a subset of the lowest entropy paths.

TABLE II

TOTAL PLANNING TIME IN SECONDS (LOWER IS BETTER) FOR THE FIRST SCENARIO; THE NUMBER OF FACTORS IN THE PRIOR GRAPH IS 12 185 AND THE NUMBER OF PATHS EVALUATED IS 2500

Method	time [s]
MP (ours)	$323.445 \pm 0.175$
rAMDЛ	$590.584 \pm 0.463$
iSAM2	$728.854 \pm 0.348$

For all scenarios the same best action was calculated. Therefore, the objective function value was identical for all methods (4243.47).

TABLE III  
COVARIANCE RECOVERY TIME

worst case [seconds]	Actual [seconds]
21.05	6.055

the *actual* is the time it takes to recover only the entries required for the states involved in the measurements for the set of all trajectories. Generally, the earlier in the robot's past trajectory the involved states are, the more of the covariance matrix entries we would need to recover, which is time consuming. However, in a scenario such as that, the time it would take to update the posterior  $R$  matrix needed for iSAM2 would be significant as well. Including the covariance recover, we can observe that our method is about 45% faster than rAMDЛ and about 65% faster than iSAM2.

Introducing a second scenario, illustrated in Fig. 16, we compare the three methods again. The basic setting is similar; the only differences are the prior mapping, start and end points, and obstacles that were added to the map. The agent infers the location of the obstacles during the prior mapping session.

Just like the previous scenario, the bounds are presented in Fig. 17, and Table IV summarizes the total planning time for each method. This scenario demonstrates a setting where there are not many observations during planning. Fig. 16 shows that most of the planning trajectories do not pass through previously mapped areas, hence not generating future measurements, which explains what we see in Table IV. Total planning time is shorter for all methods, which is caused by the small number of future measurements. We can see that our method is still the fastest without incurring any loss. However, given fewer future measurements and, as a consequence, the smaller number of Jacobian

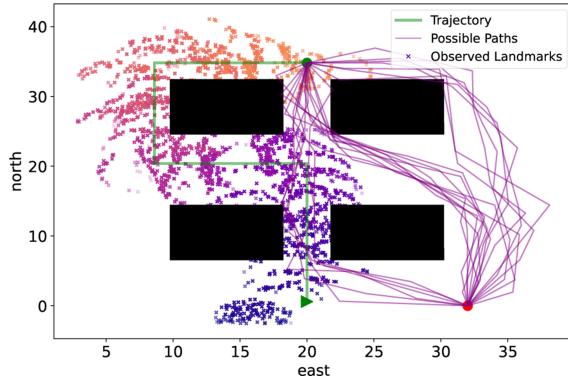


Fig. 16. Illustration of the second simulated scenario, showing a subset of the landmarks and paths; obstacles are shown in black. Each trajectory starts from the green dot and ends in the red one.

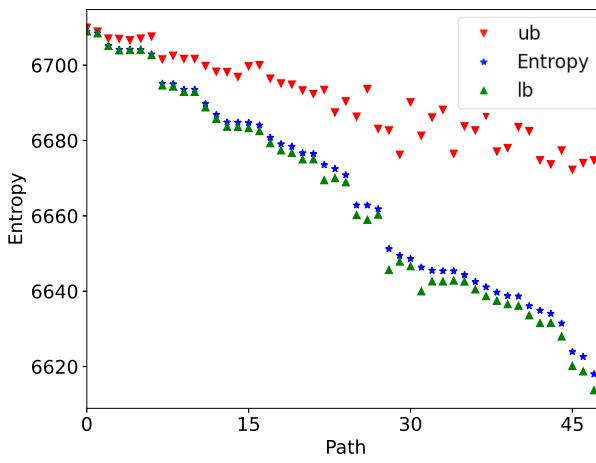


Fig. 17. Bounds for second simulated scenario, showing a subset of the lowest entropy paths.

TABLE IV  
TOTAL PLANNING TIME IN SECONDS (LOWER IS BETTER) FOR THE SECOND SCENARIO; THE NUMBER OF FACTORS IN THE PRIOR GRAPH IS 17 394 AND THE NUMBER OF PATHS EVALUATED IS 2500

Method	time [seconds]
MP (ours)	<b>41.404 ± 0.024</b>
rAMDL	50.186 ± 0.037
iSAM2	62.705 ± 0.038

For all scenarios the same best action was calculated. Therefore, the objective function value was identical for all methods (6617.20).

rows, the difference in planning time is not as drastic as in the previous scenario.

2) *Sensitivity Study*: In this section, we show the sensitivity of the different methods to the density of the prior information matrix. We use the same scenario from before, while adjusting the density of the prior information matrix by pruning factors from the prior factor graph. We use the number of landmarks connected to a certain pose (via a measurement factor) as a metric of density. This metric is then averaged across 300 evaluated paths. The results are presented in Fig. 18. We can observe that MP and rAMDL are largely unaffected by the change in prior density. This is expected since the complexity of both of them is a function of the Jacobian, as explained in Section IV-F3. On the other hand, iSAM2 is much more sensitive to the prior

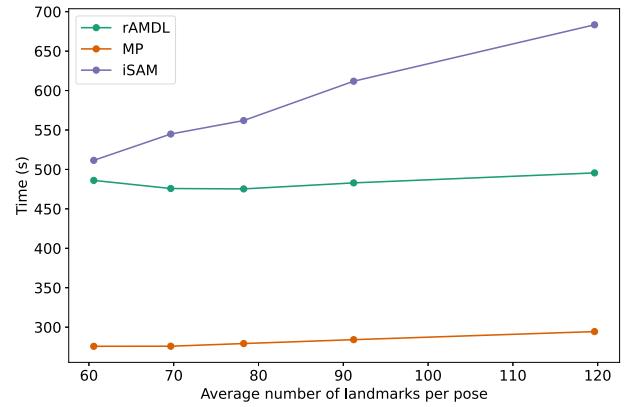


Fig. 18. Comparison of total planning time as a function of prior density.

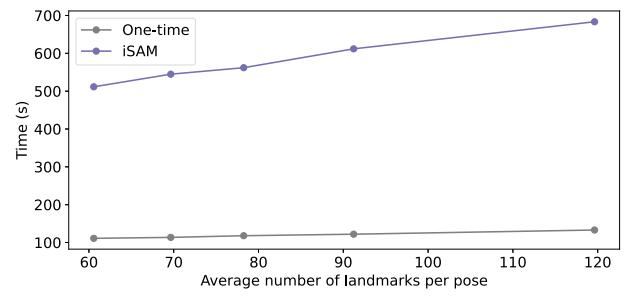


Fig. 19. Comparison of planning time as a function of prior density. iSAM2 planning time grows as a function of the prior density, while the one-time calculation needed for MP stays flat.

density as its planning time grows rapidly compared to the other methods. In order to further dissect the difference between the methods, we compare the iSAM2 planning time to the one-time calculation associated with MP and rAMDL shown in Fig. 19. The main components of the one-time are the recovery of the prior information and prior covariance matrices. Theoretically, both the one-time and iSAM2 should be affected by the change in density. In practice, we can see that the one-time calculation is affected by the change in density, but when compared to iSAM2, it effectively becomes negligible. The reason for this gap is that the one-time calculation, as its name suggests, is done only once per a given prior belief. Conversely, the components of the iSAM2 calculation that are affected by the change in density have to be evaluated from scratch for each candidate path. When comparing the total planning time of a pool of candidate paths, the growing cost associated with the one-time calculation, as the density increases, becomes negligible.

3) *Experiment*: Next, we demonstrate the use of the bounds in a real-world experiment using a DJI Robomaster S1, equipped with a ZED camera (used in a monocular configuration). GT-SAM is used for the back end while the front end is handled by SuperGlue [23]. The planning session starts after a partial mapping of an indoor environment, which is stored as a prior factor-graph. From its current pose, the robot is given a goal, it then uses the PRM to generate possible trajectories to its goal. For each method 500 paths were evaluated, each consists of 20 actions.

Fig. 20 shows the hardware setup for the experiment. Fig. 21 shows a typical feature matching on two consecutive-in-time



Fig. 20. RoboMaster S1 robot, equipped with a ZED camera.

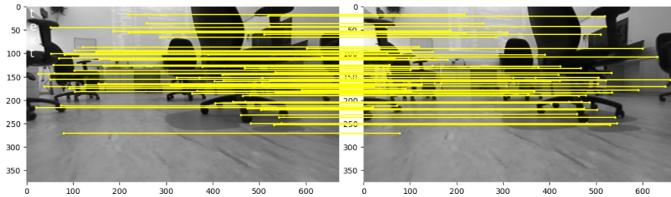


Fig. 21. SuperGlue feature matching.

TABLE V  
TOTAL PLANNING TIME IN SECONDS (LOWER IS BETTER); THE NUMBER OF FACTORS IN THE PRIOR GRAPH IS 8220 AND THE NUMBER OF PATHS EVALUATED IS 500

Method	time [seconds]
MP (ours)	<b>585.05 ± 84.78</b>
rAMDl	802.25 ± 74.13
iSAM2	1764.25 ± 17.05

For all scenarios, the same best action was calculated. Therefore, the objective function value was identical for all methods (~12 884.88).

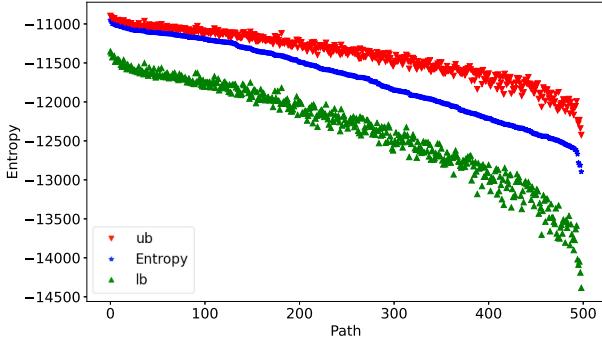


Fig. 22. Robotmaster experiment.

images. Visual odometry is used as motion model factors, and projection factors are used as observation factors.

Table V summarizes the planning time for each of the methods, and Fig. 22 shows the expected reward bounds. We can see again that our method allows for a meaningful speedup. About 70% of the paths can be pruned, while the lowest lower bound corresponds to the optimal expected reward. The planning time of our method is about 30% faster than rAMDl and about 70% faster than iSAM2. The larger difference between iSAM2 and the other two methods, compared to the same difference in Section V-C1, can be explained by the sensitivity to the density of the prior shown in the previous section. The average

number of landmarks per pose for the simulated scenario is 72.46, compared to 103.89 for the experiment.

## VI. CONCLUSION

In this article, we introduced the novel concept of observation space partitioning for BSP and POMDP planning problems. The concept is general and applies to all belief distributions and the underlying POMDP spaces. It allows for a more efficient way of identifying the optimal action by using a simplification paradigm and forming analytical bounds on the expected sum of rewards. We have demonstrated one possible use case of this concept by studying a typical active SLAM scenario with Gaussian beliefs. We have shown that both for simulated and real-world experiments, our method has a faster planning running time when compared to other SOTA methods, while identifying the same optimal trajectory.

### A. Limitations and Future Work

While the theoretical foundation of the proposed method is general, its implementation has some limitations. Tackling such limitations can form the basis for future research. The method presented is applicable for information-theoretic rewards (specifically, entropy), which are typically the computational bottleneck compared to state-dependent rewards. It is not hard, however, to extend this approach to other information-theoretic rewards such as information gain. In addition, future work can evolve in multiple different directions, which might include implementations for nonparametric belief distributions, an extension to policies over action sequences, and the possible partitioning of other POMDP spaces.

### ACKNOWLEDGMENT

The authors would like to thank Anton Gulyaev for his help with the real-world experiments and the sensitivity study.

### REFERENCES

- [1] M. Araya, O. Buffet, V. Thomas, and François Charpillet, “A POMDP extension with belief-dependent rewards,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 64–72.
- [2] M. Barenboim and V. Indelman, “Adaptive information belief space planning,” in *Proc. 31st Int. Joint Conf. Artif. Intell./25th Eur. Conf. Artif. Intell.*, 2022, pp. 4588–4596.
- [3] Y. Boers, H. Driessens, A. Bagchi, and P. Mandal, “Particle filter based entropy,” in *Proc. 13th Int. Conf. Inf. Fusion*, 2010, pp. 1–8.
- [4] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, “Generic node removal for factor-graph SLAM,” *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1371–1385, Dec. 2014.
- [5] A. J. Davison, “Active search for real-time vision,” in *Proc. Int. Conf. Comput. Vis.*, 2005, pp. 66–73.
- [6] K. Elimelech and V. Indelman, “Simplified decision making in the belief space using belief sparsification,” *Int. J. Robot. Res.*, vol. 41, no. 5, pp. 470–496, 2022.
- [7] V. Indelman, “No correlations involved: Decision making under uncertainty in a conservative sparse information space,” *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 407–414, 2016.
- [8] V. Indelman, L. Carbone, and F. Dellaert, “Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments,” *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849–882, 2015.
- [9] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artif. Intell.*, vol. 101, no. 1, pp. 99–134, 1998.

- [10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 217–236, Feb. 2012.
- [11] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.
- [12] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [13] K. Khosoussi, G. S. Shoudong, S. Huang, and G. Dissanayake, *Designing Sparse Reliable Pose-Graph SLAM: A Graph-Theoretic Approach*. Berlin, Germany: Springer, 2020, pp. 17–32.
- [14] A. Kim and R. M. Eustice, “Active visual SLAM for robotic area coverage: Theory and experiment,” *Int. J. Robot. Res.*, vol. 34, nos. 4/5, pp. 457–475, 2014.
- [15] D. Kopitkov and V. Indelman, “No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, matrix determinant lemma and re-use of calculation,” *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1088–1130, Aug. 2017.
- [16] D. Kopitkov and V. Indelman, “General purpose incremental covariance update and efficient belief space planning via factor-graph propagation action tree,” *Int. J. Robot. Res.*, vol. 38, no. 14, pp. 1644–1673, 2019.
- [17] A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies,” *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.
- [18] H. Kretzschmar and C. Stachniss, “Information-theoretic compression of pose graphs for laser-based SLAM,” *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1219–1230, 2012.
- [19] C. Papadimitriou and J. Tsitsiklis, “The complexity of Markov decision processes,” *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.
- [20] R. Platt, R. Tedrake, L. P. Kaelbling, and T. Lozano-Pérez, “Belief space planning assuming maximum likelihood observations,” in *Proc. Robot.: Sci. Syst. Conf.*, Zaragoza, Spain, 2010, pp. 587–593.
- [21] M. Popovic, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, “Online informative path planning for active classification using UAVs,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5753–5758.
- [22] N. Roy, G. J. Gordon, and S. Thrun, “Finding approximate POMDP solutions through belief compression,” *J. Artif. Intell. Res.*, vol. 23, pp. 1–40, 2005.
- [23] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperGlue: Learning feature matching with graph neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4938–4947.
- [24] T. Smith and R. Simmons, “Heuristic search value iteration for POMDPs,” in *Proc. Conf. Uncertainty Artif. Intell.*, 2004, pp. 520–527.
- [25] A. Soman, N. Ye, D. Hsu, and W. S. Lee, “DESPOT: Online POMDP planning with regularization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 1772–1780.
- [26] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using Rao-Blackwellized particle filters,” in *Proc. Robot.: Sci. Syst. Conf.*, 2005, pp. 65–72.
- [27] Z. Sunberg and M. Kochenderfer, “Online algorithms for POMDPs with continuous state, action, and observation spaces,” in *Proc. Int. Conf. Autom. Plan. Scheduling*, 2018, pp. 259–263.
- [28] O. Sztyglid and V. Indelman, “Speeding up online POMDP planning via simplification,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7174–7181.
- [29] J. Van Den Berg, S. Patil, and R. Alterovitz, “Motion planning under uncertainty using iterative local optimization in belief space,” *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [30] G. Zhang and P. A. Vela, “Good features to track for visual SLAM,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1373–1382.
- [31] A. Zhitnikov and V. Indelman, “Simplified risk aware decision making with belief dependent rewards in partially observable domains,” *Artif. Intell.*, vol. 312, 2022, Art. no. 103775.
- [32] A. Zhitnikov, O. Sztyglid, and V. Indelman, “No compromise in solution quality: Speeding up belief-dependent continuous POMDPs via adaptive multilevel simplification,” *Int. J. Robot. Res.*, early access, 2024.

**Tom Yotam** received the B.Sc. degree in mechanical engineering from Tel Aviv University, Tel Aviv, Israel, in 2017, and the M.Sc. degree in applied mathematics from the Technion, Haifa, Israel, in 2023.

**Vadim Indelman** received the B.A. and B.Sc. degrees in computer science and aerospace engineering and the Ph.D. degree in aerospace engineering from Technion, Haifa, Israel, in 2002 and 2011, respectively.

Between 2012 and 2014, he was a Postdoctoral Fellow with the Institute of Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Associate Professor with the Department of Aerospace Engineering, Technion - Israel Institute of Technology, and he is also a Member of the Technion Autonomous Systems Program (TASP), the Technion Artificial Intelligence Hub (TechAI), and the Israeli Smart Transportation Research Center (ISTRC). He is also a Member of the European Laboratory for Learning and Intelligent Systems (ELLIS). He is currently leading the Robotics vertical at TechAI, which promotes and facilitates research activities and projects within the Technion and collaboration with industry in areas related to AI and robotics. He is the Founder and Head of the Autonomous Navigation and Perception Lab (ANPL), which performs research related to single and multirobot autonomous perception, navigation and planning under uncertainty in the context of mobile robotics. His current research interests include planning under uncertainty, probabilistic inference, semantic perception, and simultaneous localization and mapping (SLAM) in single and multirobot systems.

Dr. Indelman is an Associate Editor for the *International Journal Of Robotics Research* (IJRR). He was an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS, an Editor for IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), and a Co-Chair of the IEEE Robotics and Automation Society Technical Committee on Algorithms for the Planning and Control of Robot Motion.