Simplified POMDP Algorithms with Performance Guarantees

Moran Barenboim

Simplified POMDP Algorithms with Performance Guarantees

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Moran Barenboim

Submitted to the Senate
of the Technion — Israel Institute of Technology
Sivan 5784 Haifa July 2024

This research was carried out under the supervision of Associate Professor Vadim Indelman, in the Technion Autonomous Systems Program (TASP).

The author of this thesis states that the research, including the collection, processing and presentation of data, addressing and comparing to previous research, etc., was done entirely in an honest way, as expected from scientific research that is conducted according to the ethical standards of the academic world. Also, reporting the research and its results in this thesis was done in an honest and complete manner, according to the same standards.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being:

Moran Barenboim and Vadim Indelman. Adaptive information belief space planning. In *31st International Joint Conference on Artificial Intelligence and 25th European Conference on Artificial Intelligence (IJCAI-ECAI)*, 2022.

Moran Barenboim and Vadim Indelman. Online pomdp planning with anytime deterministic guarantees. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Moran Barenboim and Vadim Indelman. Journal version of online pomdp planning with anytime deterministic guarantees. 2024. To be submitted.

Moran Barenboim, Idan Lev-Yehudi, and Vadim Indelman. Data association aware pomdp planning with hypothesis pruning performance guarantees. *IEEE Robotics and Automation Letters (RA-L)*, 10, 2023.

Moran Barenboim, Moshe Shienman, and Vadim Indelman. Monte carlo planning in hybrid belief pomdps. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):6827–6834, 2023.

Idan Lev-Yehudi, Moran Barenboim, and Vadim Indelman. Simplifying complex observation models in continuous pomdp planning with probabilistic guarantees and practice. In *38th AAAI Conference on Artificial Intelligence (AAAI-24)*, 2024.

Acknowledgements

I am deeply grateful to my supervisor, Professor Vadim Indelman, for his guidance and high standards, which have been essential to my development as a researcher.

I also appreciate the Technion Autonomous Systems Program (TASP) and the Autonomous Navigation and Perception Lab (ANPL) for creating a supportive environment.

Finally, I want to thank my wife, Lior, whose support made this journey possible.

The generous financial help of the Technion is gratefully acknowledged.

Contents

Abstract

Autonomous agents operating in real-world scenarios frequently encounter uncertainty and make decisions based on incomplete information. This challenge can be structured mathematically through the lens of partially observable Markov decision processes (POMDPs). While POMDPs offer a robust framework for planning under uncertainty, finding an optimal plan for a POMDP can be computationally intensive and is feasible only for simpler tasks. In response, the last two decades have witnessed the rise of approximate algorithms, like tree search and sample-based approaches, as leading solutions for tackling more complex POMDP problems. Despite their effectiveness, these algorithms typically offer only probabilistic guarantees or, in some cases, no formal guarantees at all.

In our research, we have focused on addressing these limitations by developing a range of simplified algorithms with formal, deterministic guarantees. These simplified algorithms operate on a selected subset of the state and observation spaces, commonly considered in state-of-the-art algorithms, while providing mathematical guarantees and computational efficiencies compared to the non-simplified algorithm. Initially, we focused on a belief-dependent reward framework, simplifying the reward calculation by narrowing down the observation space. Then, we have applied a simplification to the state space in the context of hybrid-belief and data-association aware POMDPs, which otherwise may grow exponentially. Ultimately, we extended our approach to a broad POMDP framework, simplifying both state and observation spaces, and providing deterministic guarantees with respect to the optimal solution.

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

The focus of our research lies in the area of decision-making, specifically decision-making under uncertainty. The primary objective of a decision-making solver is to identify an optimal plan for a given sequential problem. Uncertainty arises from the limited knowledge that an operating agent has about its environment. This issue is prevalent in many practical autonomous systems, such as autonomous vehicles, industrial manipulators on conveyors, and drones navigating crowded spaces. The uncertainty is due to incomplete information about the environment and the current state within it. For example, a robotic platform may rely on various sensors to perceive its surroundings, but these sensors typically have limited range, are prone to noise, and can be affected by occlusions. Additionally, uncertainty can stem from modeling approximations that translate the real world into mathematical or programmatic representations.

The goal of this work is to develop efficient algorithms for online decision-making under uncertainty while ensuring performance guarantees.

1.2 Preliminaries

Decision-making under uncertainty can be approached in various ways, but one of the most commonly accepted and theoretically sound methods is to formulate the problem as a partially observable Markov decision process (POMDP).

In a POMDP, the state encapsulates all the relevant information about the environment, including the agent's pose, the map of the environment, the temperature, and more. However, the agent does not have direct access to this state. Instead, it receives observations such as sensor measurements, which may provide only partial or noisy information about the state. By utilizing accumulated knowledge over time, including all past observations and actions, the agent can infer information about the actual state. One method for acquiring more information about the state is Bayesian inference. In Bayesian inference, the agent maintains a prior distribution over all possible states and

updates this distribution as new information arrives, forming a posterior distribution. This distribution over the states is also known as the belief.

Given an inference mechanism, the agent can perform future planning by assuming different courses of actions and observations, and projecting various trajectories into the future. This form of planning is known as tree search planning. One of the main benefits of tree search planning is its increased efficiency, achieved by focusing only on the relevant future beliefs that the agent may encounter. By projecting different futures and scoring them, the agent can choose the action that, in expectation, leads to the best outcome.

We begin by formally introducing the POMDP and its associated notations.

1.2.1 POMDP Notations

POMDP is defined as a 7-tuple, $M = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, R, b_0)$, where, $\mathcal{X}, \mathcal{A}, \mathcal{Z}$ denote the state, action and observation spaces, respectively. $T(x', x, a) = \mathbb{P}(x' \mid x, a)$ represents the transition function, such that given the current state $x \in \mathcal{X}$ and action $a \in \mathcal{A}$ returns the likelihood of the next state, $x' \in \mathcal{X}$. The observation function, $O(z, x) = \mathbb{P}(z \mid x)$ returns the likelihood of obtaining observation $z \in \mathcal{Z}$ given the state $x \in \mathcal{X}$. The prior, b_0, denotes the initial belief distribution, which returns the likelihood of being in state $x \in \mathcal{X}$ at time step $t = 0$, $b_0(x) = \mathbb{P}_0(x)$. Last, $R(b, a)$ is a reward function, which given a belief and an action, returns a scalar value for scoring being in belief b and performing action a.

Additionally, we use history, H_t, as a shorthand for all past action-observation sequences $H_t = \{a_0, z_1, a_1, z_2, \ldots, a_{t-1}, z_t\}$ up to time t. We use H_t^- to denote the same history, without the last observation, i.e. $H_t^- = (a_0, z_1, \ldots, a_{t-1})$. The likelihood of state x given a history H_t is denoted as the belief, $b_t(x) = \mathbb{P}(x \mid H_t)$, and $b_t^-(x) = \mathbb{P}(x \mid H_t^-)$ for a belief conditioned on H_t^-.

1.2.2 Inference

As observations provide only partial information about the state, the true state of the agent is unknown. Therefore, the agent maintains a probability distribution function over the state space, also known as a belief. At each time step t the belief is being updated according to Bayes rule, using the transition and observation models. Given the performed action a_{t-1} and the received observation z_t, the belief update performed according to,

$$b_t(x) = \eta_t \int_{x_{t-1} \in X} \mathbb{P}(z_t|x)\mathbb{P}(x|x_{t-1}, a_{t-1}) b_{t-1}(x_{t-1})\, dx_{t-1} \tag{1.1}$$

where η_t is a normalization constant,

$$\eta_t = \int_{z_t \in Z} \int_{x_{t-1} \in X} \mathbb{P}(z_t|x)\mathbb{P}(x|x_{t-1}, a_{t-1}) b_{t-1}(x_{t-1})\, dx_{t-1} dz_t. \tag{1.2}$$

However, performing inference is generally intractable due to integration over arbitrarily-shaped distribution functions. Even for discrete spaces this calculation becomes prohibitively expensive for large state and observation spaces. In the following sections we will cover different approximation methods that aim to relax that, some of which are particularly appealing for planning and decision making.

1.2.3 Planning

To solve a POMDP optimally, the agent needs to find a plan that maximizes some objective function, usually defined as the sum of expected future reward values over the unknown states. To do so, it is required to reason about all possible future actions and their resulting observations.

A policy, $\pi(\cdot)$, is a function that realizes a specific plan and maps every belief to an action to be executed. Each plan, or policy, is evaluated according to a value function. Given a belief at time t, each policy corresponds to a value function,

$$V^{\pi}(b_t) = \mathbb{E}_z \left[\sum_{\tau=t}^{\mathcal{T}-1} R(b_\tau, \pi_\tau(b_\tau), b_{\tau+1}) \right], \tag{1.3}$$

which is the expected cumulative reward following the policy, π. Similarly, an action-value function,

$$Q^{\pi}(b_t, a_t) = \mathbb{E}_{z_{t+1}} \left[R(b_t, a_t, b_{t+1}) + V^{\pi}(b_{t+1}) \right], \tag{1.4}$$

is the value of executing action a_t in b_t and then following the policy π. The optimal policy, $\pi^{\star}$, is a policy that maximizes the value function, $V^{\pi^{\star}}(b_t) = \max_{\pi} V^{\pi}(b_t)$.

The scope of the POMDP formulation is quite flexible, and depending on the problem at hand, the horizon of the future actions and observations may be finite or infinite, the state, observation, and action spaces may be discrete or continuous and the belief over the state space may be structured (e.g. Gaussian) or arbitrarily distributed. Unfortunately, finding the optimal solution of a POMDP is intractable to all but small problems [39].

1.3 Contribution

With the background set, we are now equipped to discuss the contributions presented in this dissertion. Due to the intractability of planning and eventually arriving at the optimal plan we are forced to consider an approximate solution. In this thesis, we simplify the required calculations for finding a good plan, albeit mostly not optimal, while grounding the resulting solution with a mathematical relationship to the underlying, intractable solution.

1.3.1 Adaptive Information Belief Space Planning - Simplifying observation spaces

We begin by considering a specific aspect of planning under uncertainty, named Belief Space Planning or ρ-POMDPs. In this line of work, we consider problems that target uncertainty reduction as their optimization criteria. Using uncertainty reduction as the optimization criteria, requires a dedicated reward function for this task. Common reward functions that aim the reduce the uncertainty grasp some property of the belief distribution. However, these functions are usually more computationally challenging than reward functions in standard POMDPs. More concretely, we consider entropy as part of the reward function and show that it adds significant computational burden. Then, to alleviate some of the computational cost, we perform simplification over the observation space and relate the solution to the underlying, computationally challenging POMDP. We show that not only we can reduce computational time, but can also guarantee the same policy as the more expensive POMDP.

1.3.2 Discrete-Continuous State Spaces - Simplifying state spaces

In this line of work, we consider state spaces that involve both discrete and continuous states. We show that such cases may lead to computational complexity that is significantly worse than standard POMDPs. Specifically, we consider cases where the belief may be represented either as a mixture or as a hybrid distribution. To alleviate some of the computational burden, we suggest new approximate algorithms that consider only a subset of the mixture-or-hybrid components and derive theorems to link between the simplified and the original representation of the belief.

1.3.3 Online POMDPs with Deterministic Guarantees - Simplifying observation and state spaces

Last, we consider POMDPs with just discrete spaces. We derive, for the first time, deterministic guarantees that link any solution obtained throughout online planning phase to the optimal solution. We show that this relationship, described as lower and upper bounds on the difference between the approximate solution and the optimal one, can be used on most state-of-the-art algorithms with mild impact on the computational complexity. Then, we demonstrate how to utilize the bounds for the decision-making phase, and show that in some cases it can result in an improved overall performance. Last, we show that apart from using the theorem for the final decision-making, the theorem benefits from few appealing properties and can also be used for the planning phase. Last, we demonstrate that using the deterministic bound for planning and decision-making may further improve the overall performance of the solution.

1.4 Related Work

Due to the intractability of finding an exact solution to a POMDP, various approximate algorithms have been developed. Tree search algorithms, which are the main focus of this dissertation, are a prominent approach for such approximations. Instead of considering all possible belief states, tree search algorithms reduce the belief space to a reachable subset, starting from the prior belief node. In online tree-search algorithms, rather than calculating a policy in advance, the planner needs to find an approximately optimal action by building a tree at every time step.

1.4.1 Monte Carlo Tree Search and POMCP

Monte Carlo Tree Search (MCTS) [6] is a heuristic search algorithm for decision processes, especially useful in large state spaces. MCTS builds a search tree incrementally, guided by random simulations (rollouts). The algorithm balances exploration and exploitation using the Upper Confidence Bounds for Trees (UCT) algorithm [27], which selects actions that maximize an upper confidence bound on the estimated value.

Partially Observable Monte Carlo Planning (POMCP) [45] extends MCTS to the POMDP framework by utilizing particle filtering to represent beliefs as sets of state particles. During each rollout, a single state particle is recursively propagated from the root node to the leaves of the tree. POMCP handles large state and observation spaces by aggregating Monte Carlo rollouts of future scenarios in a tree structure. It adaptively trades off between actions that lead to unexplored areas of the tree and actions that lead to rewarding areas by utilizing UCT.

While POMCP is considered a state-of-the-art algorithm, it is limited to discrete state, action, and observation spaces.

1.4.2 DESPOT and Its Variants

The Determinized Sparse Partially Observable Tree (DESPOT) algorithm [46] is another notable approximate solver for POMDPs. DESPOT performs forward search over a sparse subset of the belief space by sampling a set of representative scenarios and building a sparse belief tree over these scenarios. It relies on branch-and-bound techniques to prune suboptimal actions and uses dynamic programming to update the value function estimates at each node.

Anytime Regularized DESPOT (AR-DESPOT) [57] is a variant of DESPOT that introduces regularization to handle the exploration-exploitation trade-off more effectively. Similar to POMCP, AR-DESPOT performs forward search and propagates a single particle from the root node down to its leaves. It uses a branch-and-bound approach in the forward search and utilizes dynamic programming techniques to update the value function estimate at each node.

DESPOT and its descendants consider α-vectors in their derivations and are thus

limited to reward functions that are linear in the belief. However, information-theoretic functions are usually not linear with respect to the belief and thus not supported by these approaches.

1.4.3 Progressive Widening and PFT-DPW

To handle continuous action and observation spaces, Sunberg and Kochenderfer [48] proposed the POMCPOW algorithm, which extends POMCP by incorporating Progressive Widening (PW) [8]. PW allows the algorithm to handle continuous spaces by limiting the branching factor of the tree. Actions and observations are sampled according to a power-law distribution, ensuring that the tree remains tractable.

The Partially Observable Monte Carlo Planning with Double Progressive Widening (PFT-DPW) algorithm [48] further extends this idea. In PFT-DPW, each expanded node in the belief tree contains the same number of particles and is better suited for belief-dependent rewards. By maintaining a fixed number of particles at each node, PFT-DPW provides a more accurate approximation of the belief, which is important for problems with belief-dependent rewards.

However, due to the exploratory nature of these algorithms, most belief nodes either contain only a few particles or contain just a small number of observation branches, making them less suitable for approximating belief-dependent rewards.

1.4.4 Belief-Dependent Reward Functions

A reward in a POMDP is commonly a function that receives a state and action as input and maps each state to a scalar value. However, when the reward receives a belief as input, the problem formulation is considered an extension to POMDP, called ρ-POMDP [1], POMDP-IR [47], or Belief Space Planning (BSP) [41], [54], [21]. An objective function on the distribution itself allows reasoning about the uncertainty directly, which arises in many problems such as active localization [7], information gathering [17], and active SLAM [26]. Although the POMDP formulation allows reasoning about uncertainty implicitly, it is insufficient for problems where the goal is defined over a distribution. For instance, in the active localization problem, the goal of the agent is not to reach a certain destination but to reduce uncertainty about its state.

Common approaches for measuring uncertainty are information-theoretic functions, such as entropy, information gain, and mutual information. For continuous distributions, calculating the exact values of information-theoretic rewards involves intractable integrals in the general case. Thus, they are amenable to different approximations, such as kernel density estimation (KDE), Voronoi diagrams [34], or sampling-based approximations [4]. Unfortunately, all such approximations are expensive to compute, as they require quadratic costs in the number of samples and are usually the bottleneck of planning algorithms. Even when the belief is described as a structured multivariate Gaussian, exact computation becomes expensive as the dimension of the state grows

[28].

More closely related to the work presented in this thesis is [52], which interleaves MCTS with ρ-POMDP. Their approach considers a discrete observation space. In each traversal of the tree, their algorithm adds a fixed set of particles propagated from the root node, which results in an increased number of samples at each node as the algorithm progresses. According to the authors, the main motivation is a reduced asymptotic bias. Given a time budget, a significant reduction in the number of nodes explored is observed by the authors of [52], which in turn impacts the quality of the policy. [14] consider a belief-dependent reward, in which they build upon PFT-DPW [48]. Instead of maintaining the same particle set in each posterior node, they reinvigorate particles in every traversal of the posterior node. Then, they propose to average over different estimations of the reward function. [14] suggest estimating the reward function using KDE, which scales quadratically with the number of state samples. [49] consider a sampling-based approximation to evaluate differential entropy. They propose a simplification procedure that alters the number of state samples and prunes sub-optimal action branches using bounds relative to the non-simplified estimator. In a follow-up work, [50] propose an approach to interleave simplification with MCTS while maintaining tree consistency, thereby increasing computational efficiency.

1.4.5 Planning with Mixed Discrete-Continuous State Spaces

In general, all random variables in a hybrid belief are coupled, and the number of hypotheses, i.e., realizations of discrete variables, may be combinatorially large with the number of ambiguous objects and classes or even develop exponentially with time given ambiguous data associations. Therefore, without resorting to approximation, the size of the considered belief quickly becomes prohibitively large, and the computational complexity of the corresponding problem becomes impossible to handle.

The research community has been extensively investigating passive inference approaches where the considered belief is hybrid. In [42], the authors proposed a message-passing algorithm to correctly identify loop closures by optimizing a hybrid factor graph [29]. A convex relaxation approach over a discrete-continuous graphical model was presented in [31] to capture perceptual aliasing and find the maximal subset of internally coherent measurements, i.e., correct data association.

In a semantic Simultaneous Localization and Mapping (SLAM) problem, the coupling between poses and discrete object class labels can be used both for disambiguating data association and pose estimation. [51] presented a recursive Bayesian approach for localization and semantic mapping in ambiguous environments using a hybrid belief over camera and object poses, with classification and data association hypotheses. In [10], the authors utilized the mm-iSAM model from [15] to solve a semantic non-Gaussian SLAM problem with unknown data associations, using non-parametric belief propagation, while in [11], the authors used the max-mixture model [38] to solve a max-

imum a-posteriori inference designed specifically for the nonlinear Gaussian case. In [35], the authors proposed a sampling-based approach with expectation-maximization (EM) that used the most likely class semantic measurements to perform batch inference, while in [5], the most likely class and bounding box measurements were used, in addition to geometric measurements, to perform SLAM and data association disambiguation with EM as well.

In spite of the significant progress made within the SLAM community, such a hybrid setting has received scant attention from the planning community. As such, most off-the-shelf, state-of-the-art POMDP online solvers do not directly support hybrid beliefs. Specifically, [45] introduced POMCP, an adaptation to Monte Carlo Tree Search (MCTS) for POMDPs using the UCT algorithm [27] to guide the action selection process. POMCPOW and DESPOT [48, 46] employ transition and observation models to efficiently propagate particles from the prior belief as an efficient approximation for belief update. However, in the context of hybrid beliefs, the belief update may not be as efficient since it would require knowledge of the hypotheses' probabilities, which are not presumed to be given.

POMDPs can also be converted into belief Markov decision processes (BMDPs) to utilize MDP solvers. PFT-DPW [48] and AI-BSP [2] are two such solvers, where belief-states replace states in the original MDP algorithms. However, performing inference with hybrid belief is hardly efficient due to a large number of hypotheses. For instance, in ambiguous data association scenarios, the number of hypotheses grows exponentially with time, making full inference intractable.

Sequential Monte Carlo (SMC) methods, also known as particle filters, are non-parametric inference mechanisms that use sampling to approximate the posterior belief. Using particle filters, the belief at each time step is represented by K samples, sampled from the known models. Particle filters are efficient in terms of time complexity, which is linear in the number of state particles. Furthermore, the structure of SMC methods allows anytime inference to allow better posterior representation given more time.

This property is commonly exploited in many state-of-the-art (SOTA) planning algorithms, such as [45, 46, 48] and [30]. However, for both inference [53] and planning [45, 46, 48, 30], SMC methods are usually used in the filtering paradigm, which is susceptible to particle depletion and over-confident belief representation.

Optimization inference methods are the current SOTA in passive SLAM, e.g., [23]. These methods use smoothing paradigms that keep track of an increasing number of state variables, significantly outperforming state-dimensionality of particle filters. [23] represents the SLAM instance as a Bayes tree, then utilizes the structure of the joint probability distribution to incrementally update the posterior belief, resulting in a real-time, efficient inference engine. Such methods were also used in planning, e.g., [21, 36], allowing long-term reasoning about loop closures and uncertainty reduction. However, smoothing inference methods are usually limited to Gaussian distributions, unable to reason about multi-modal distributions commonly arising in, e.g., ambiguous

data associations in SLAM instances, different object class hypotheses, and multi-modal transition models.

Multi-modal inference has recently received increased attention from the SLAM community. [15] uses kernel density estimation (KDE) with Gaussian kernels to represent nonparametric belief distributions. Posterior update is done using Gibbs sampling while exploiting the joint distribution structure, inspired by iSAM2 [23]. [18] uses a tree structure to represent multiple belief hypotheses, sharing past calculations among hypotheses that share partial histories. To keep calculations tractable, a pruning algorithm is proposed. [20] utilizes normalizing flows from deep learning to transform arbitrary distributions to a normal distribution, solving SLAM problems with non-Gaussian factors. However, current reported results yet to scale towards high-dimensional problems in real-time.

While addressing the challenge of ambiguous data associations (DA) has been extensively researched in the passive inference community, the planning community has had relatively few attempts at supporting ambiguous DA. General state-of-the-art POMDP planners, such as DESPOT, POMCPOW, or PFT-DPW [48, 46] do not directly support DA out-of-the-box. Although they can be altered to support DA, e.g., by replacing the observation model with a mixture of observation models, an ad-hoc variation will often result in particle depletion due to the multi-modal nature of multiple hypotheses belief. Particle depletion results in an overconfident and potentially incorrect action selection due to the low representation of likely state particles in a belief.

A more dedicated approach for handling ambiguous DA could be to explicitly maintain multiple representations of conditional beliefs, each depending on different DA histories. A naive attempt to perform planning with all hypotheses results in an exponentially increasing number of hypotheses, which is computationally infeasible. Instead, the authors of [40] introduced DA-BSP, which allows reasoning about future data association hypotheses within a belief space planning framework for the first time. [43] suggested reducing the computational complexity of DA-BSP by selecting only a small subset of hypotheses and providing bounds over the loss in solution quality. [43] was later extended to a non-myopic setting in [44]. The ARAS framework proposed in [19] leveraged the graphical model presented in [18] to reason about ambiguous data association in future beliefs using multi-modal factors to model discrete ambiguities. Due to its high computational burden, these approaches did not aim at closed-loop POMDP planning, neglecting its mathematical soundness.

1.4.6 Online POMDP Planning with Anytime Deterministic Guarantees

As previously mentioned, a prominent search algorithm addressing the challenges posed by large state and observation spaces in POMDPs is POMCP [45]. From the solution quality standpoint, the mathematical guarantees on the provided solution by POMCP

are asymptotic, and the quality of the solution remains unknown within any finite time frame.

In contrast to POMCP, Regularized DESPOT offers a probabilistic lower bound on the value function obtained at the root node, providing theoretical appeal by measuring its proximity to the optimal policy.

As previously mentioned, POMDPs serve as a comprehensive mathematical framework for addressing uncertain sequential decision-making problems. Despite their applicability, most problems framed as POMDPs struggle to achieve optimal solutions, largely due to factors such as large state spaces and an extensive range of potential future scenarios. The latter tends to grow exponentially with the horizon, rendering the solution process computationally prohibitive.

Few more notable approaches for solving POMDPs with either discrete or continuous spaces include POMCPOW [48], LABECOP [16], and AdaOPS [56], which leverage explicit use of observation models. These algorithms employ importance sampling mechanisms to weigh each state sample based on its likelihood value, which is assumed to be known. Although these methods have exhibited promising performance in practical scenarios, they currently lack formal guarantees. To address this gap, [33, 32] introduced a simplified solver aimed at bridging the theoretical gap between the empirical success of these algorithms and the absence of theoretical guarantees for continuous observation spaces. In [32], probabilistic guarantees were derived for the simplified solver concerning its proximity to the optimal value function, thus contributing to a more comprehensive understanding of POMDP planning in both discrete and continuous settings.

Chapter 2

Adaptive Information Belief Space Planning

In this chapter, we show an approach to alleviate the computational burden of calculating reward at each belief node of the tree, while guaranteeing an identical solution. We focus on Shannon's entropy and differential entropy, alongside state-dependent reward functions. Our approach relies on clustering different nodes and evaluating an approximated belief-dependent reward once on this entire cluster, that is, all the nodes within a cluster share the same reward value, see figure 2.1. As a result, the estimated value function is affected. To relate the approximated value function to the one that would originally be calculated, each node maintains a lower and upper bound on the value function.

Our main contributions are as follows. First, we introduce an abstract observation model. We use the model to form an abstraction of the expected reward, namely, a weighted average of state-dependent reward and entropy. Then, using the abstract model, we show how computational effort is alleviated. Second, we derive deterministic lower and upper bounds for the underlying expected reward values and the value function. Third, we introduce a new algorithm, which is able to tighten the bounds upon demand, such that the selected action is guaranteed to be identical to the non-simplified algorithm. Last, we evaluate our algorithm in an online planning setting and show that our algorithm outperforms the current state-of-the-art.

2.1 Preliminaries

We define POMDP as defined in section 1.2.1. Additionally, we focus on a reward function defined as a weighted sum of state-dependent reward and entropy,

$$R(b, a, b') = \omega_1 \mathbb{E}_{x\sim b'}\left[r_x(x, a)\right] + \omega_2 \mathcal{H}(b'), \tag{2.1}$$

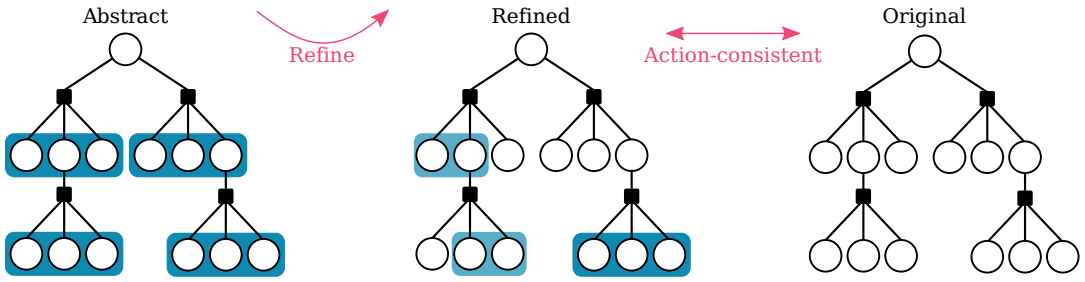


Figure 2.1: An illustration of our approach. The blue clusters correspond to a single evaluation of the reward function across different posterior nodes, which is faster to compute. Then, the algorithm initiates a refinement procedure; the refined clusters guarantee the same action selection as the original reward evaluation.

where b' is the subsequent belief to b and $\mathcal{H}(\cdot)$ is either differential entropy or Shannon's entropy. The dependence of (2.1) on both b and b' stems from the definition of the differential estimator, as will be shown in Section 2.2.2. Given a belief at time t, each policy corresponds to a value function,

$$V^{\pi}(b_t) = \mathbb{E}_z \left[\sum_{\tau=t}^{\mathcal{T}-1} R(b_\tau, \pi_\tau(b_\tau), b_{\tau+1}) \right], \tag{2.2}$$

which is the expected cumulative reward following the policy, π. Similarly, an action-value function,

$$Q^{\pi}(b_t, a_t) = \mathbb{E}_{z_{t+1}} \left[R(b_t, a_t, b_{t+1}) + V^{\pi}(b_{t+1}) \right], \tag{2.3}$$

is the value of executing action a_t in b_t and then following the policy π.

2.1.1 Belief-MDP

A belief-MDP is an augmentation of POMDP to an equivalent MDP, by treating the belief-states in a POMDP as states in the Belief-MDP. Subsequently, algorithms developed originally for MDPs can be used for solving POMDPs or BSP problems with slight modifications, a property that we exploit in this chapter.

2.2 Expected Reward Abstraction

In this section we introduce the notion of an abstract observation model and show how this model can be utilized to ease the computational effort. We then derive bounds on the expected reward and, as a consequence, on the value function.

We start this section with a theoretical derivation, where we assume the reward can be calculated analytically. This appears in special cases, e.g. when the reward is solely entropy and the belief is parametrized as a Gaussian or when the state space is discrete. In this part we assume the observation space is discrete and thus the expected reward can be calculated analytically. In the second part of this section, we relax those assumptions. Generally, the observation and state spaces can be continuous and the

belief may be arbitrarily distributed. To deal with such cases, we derive an estimator of the expected reward, in which the belief is approximated by utilizing a particle filter.

For both the discrete and continuous observation spaces, we present an abstract observation model,

Definition 2.2.1 (Abstract observation model)**.** An abstract observation model assigns a uniform probability to all observations within a single set,

$$\bar{O}(z^j \mid x) \doteq \frac{1}{K} \sum_{k=1}^{K} O\left(z^k \mid x\right) \qquad \forall j \in [1, K] \tag{2.4}$$

where $O\left(z^k \mid x\right)$ corresponds to the original observation model over different observation realizations, z^k, and K denotes the cardinality of observations within that set.

The abstract model aggregates K different observations and replaces the original observation model when evaluating the reward, see Figure 2.2. In the continuous case we revert to observation samples, thus the summation in (2.4) corresponds to different observation samples. A more precise explanation of the continuous case will be given in Section 2.2.2.

2.2.1 Discrete Observation Space

Since calculating the exact value of a reward function in every belief node is expensive, we now formulate an approach to evaluate rewards once for an entire set of K posterior beliefs. Such an abstraction results in a decreased number of reward evaluations in planning. We show that when constructing the aggregation scheme as a uniform distribution over a set of observations, one can achieve tight upper and lower bounds on the expected entropy, defined as $-\mathbb{E}\left[\sum_x b(x) log(b(x))\right]$. Moreover, we show that abstraction for the expected state-dependent reward does not affect its value, which remains identical with and without abstraction.

Denote the cardinality of observation space with N_o, we partition the observations to C clusters and denote the number of observations within each cluster as K, see Figure 2.2. Generally, each cluster may contain a different number of observations, $N_z = \sum_{c=1}^{C} K(c)$; For clarity, we assume K is identical for all clusters, but the results below are easily extended to the more general case.

Our *key result*, stated in the lemma below, corroborates the intuition that utilizing an abstract observation model results in a reduced number of reward evaluations.

Lemma 2.2.2. *Evaluation of the expected reward with an abstract observation model,* (2.4)*, requires only C evaluations of the reward, instead of N_o, where $C = \frac{N_o}{K}$. That is,*

$$\begin{aligned} &\sum_{n=1}^{N_z} \bar{\mathbb{P}}\left(z_{t+1}^n \mid H_{t+1}^-\right) R\left(b_t, a_t, \bar{b}_{t+1}\right) = \\ &K \sum_{c=1}^{C} \bar{\mathbb{P}}\left(z_{t+1}^{c \cdot K} \mid H_{t+1}^-\right) R\left(b_t, a_t, \bar{b}_{t+1}\right). \end{aligned} \tag{2.5}$$

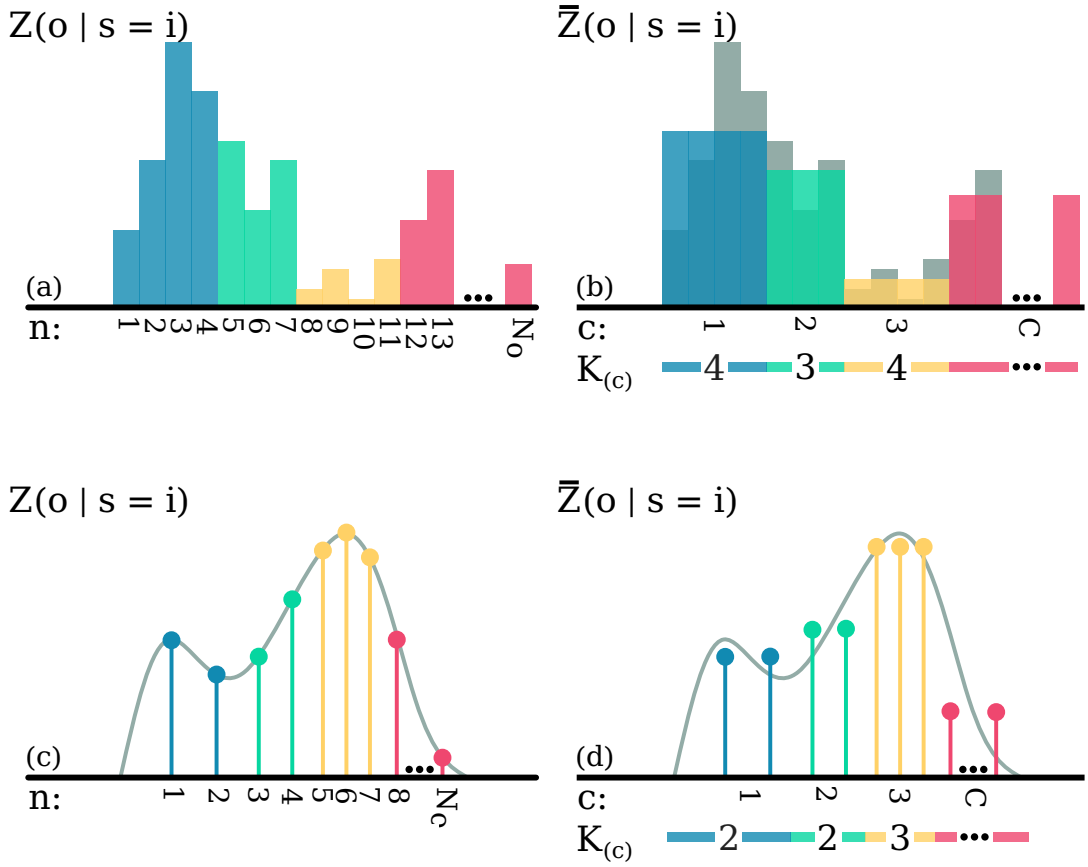


Figure 2.2: An abstraction of the observation model: (a) original discrete observation model with N_o observations. (b) Abstract discrete observation model with C clusters. (c) Sample set from the original continuous observation model, with N_o observations. (d) Abstract sample set with C clusters.

Proof. see appendix A.1. ■

Here, $z_{t+1}^{c \cdot K}$ denotes a single representative observation of the cluster c and

$$\bar{\mathbb{P}}\left(z^n \mid H^-\right) \doteq \sum_{x \in S} \bar{O}(z^n \mid x) b^-, \tag{2.6}$$

$$\bar{b} \doteq \frac{\bar{O}(z^n \mid x) b^-}{\sum_{x' \in S} \bar{O}(z^n \mid x') b^-} \,. \tag{2.7}$$

For the continuous state case, simply replace summations with integrals. Furthermore, the expected state-dependent reward remains unchanged when evaluated over abstract belief and expectation,

Lemma 2.2.3. *The value of the expected state-dependent reward is not affected by the abstraction shown in* (2.4)*, i.e.,*

$$\mathbb{E}_o \left[\mathbb{E}_{x \sim b}\left[r_x\left(x, a\right)\right]\right] = \bar{\mathbb{E}}_o \left[\mathbb{E}_{x \sim \bar{b}}\left[r_x\left(x, a\right)\right]\right]. \tag{2.8}$$

Proof. see appendix A.2. ■

Note that $\{\mathbb{E}_o, \mathbb{E}_b\}$ and $\{\bar{\mathbb{E}}_o, \mathbb{E}_{\bar{b}}\}$ correspond to expectations with the original and abstract observation models, (2.6) and (2.7).

We now transition to the main theorem of this chapter. Using (2.4) as the abstraction mechanism, we show that,

Theorem 2.1. *The expected entropy is bounded from above and below by,*

$$0 \leq \bar{\mathbb{E}}_z \left[\mathcal{H}\left(\bar{b}\right)\right] - \mathbb{E}_z \left[\mathcal{H}\left(b\right)\right] \leq log(K). \tag{2.9}$$

Proof. see appendix A.3. ■

A direct implication of this result is a bounded sub-optimality, which we state explicitly in corollary 2.2, while increasing computational efficiency by a factor of K. The bounds hold in the worst-case sense, i.e. regardless of the choice of which observations one chooses to cluster together. Note that the difference between the expected entropy and the abstracted one is bounded from below by zero. Since the entropy evaluates the uncertainty, the interpretation of this result is quite intuitive; the uncertainty cannot reduce when using abstracted models. The upper bound depends on the number of observations we choose to abstract, K. When $K = 1$, that is, each cluster contains a single observation, both the upper and lower bounds are zero and the abstract expected entropy equals the original expected entropy. From (2.1), (2.8) and (2.9) it follows that,

$$0 \leq \bar{\mathbb{E}}_z \left[R\left(b, a, \bar{b}'\right)\right] - \mathbb{E}_z \left[R\left(b, a, b'\right)\right] \leq \omega_2 log(K). \tag{2.10}$$

We now generalize those results and show that the value function is bounded. An abstract value function is defined as,

$$\bar{V}^{\pi}(b_t) = \bar{\mathbb{E}}_{z_{t+1}} \left[R(b_t, \pi_t(b_t), \bar{b}_{t+1})\right] + \mathbb{E}_{z_{t+1}} \left[\bar{V}^{\pi}(b_{t+1})\right]. \tag{2.11}$$

As a direct consequence of (2.10) and (2.11),

Corollary 2.2. *The difference between the original value function and the abstract value function is bounded by,*

$$0 \leq \bar{V}^{\pi}(b_t) - V^{\pi}(b_t) \leq \mathcal{T} \cdot \omega_2 log(K). \tag{2.12}$$

Proof. see appendix A.4. ■

This result allows us to bound the loss when applying observation abstraction. In Section 2.3, we devise an algorithm that adapts the bounds so that the same best action will be chosen, with and without abstraction, while expediting planning time.

2.2.2 Continuous Observation Space

Planning with entropy as reward over a continuous observation space is more cumbersome as it requires calculation of,

$$\mathbb{E}[\mathcal{H}(b_t)] = -\int_{z_t} \mathbb{P}(z_t \mid H_t^-) \int_{x_t} b(x_t) log(b(x_t)). \tag{2.13}$$

First, the probability density function of the belief may be arbitrary, so the integral over the state has no closed-form expression. Moreover, usually, there is no access to the density functions due to the difficulty of exact Bayesian inference, but only to samples. Second, even if the differential entropy could be evaluated, integrating over the observation space makes this calculation intractable. Consequently, it is only feasible to estimate the value function, here denoted by $\hat{V}^{\pi}(\cdot)$. To approximate the posterior belief at each time step, we employ the commonly used particle filter, see e.g. [53]. Inspired by derivations in [4], we estimate (2.13) using state samples obtained via a particle filter. We then sample observations using the given observation model, conditioned on the state particles. This is a common procedure in tree search planning, see for example [48]. Using the state and observation samples, we derive an estimator to (2.13),

$$\hat{\mathbb{E}}[\mathcal{H}(\hat{b}_t)] = -\hat{\eta}_t \sum_{m=1}^{M} \sum_{i=1}^{N} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i \cdot \tag{2.14}$$
$$log\left(\frac{O\left(z_t^m \mid x_t^i\right) \sum_{j=1}^{N} T\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j}{\sum_{i'=1}^{N} O\left(z_t^m \mid x_t^{i'}\right) q_{t-1}^{i'}}\right),$$

where $\hat{b} \doteq \{q^i, x^i\}_{i=1}^N$ denotes the belief particles with weights q^i; M, N are the number of observation and state samples accordingly and $\hat{\eta}_t = \frac{1}{\sum_{m=1}^{M} \sum_{i=1}^{N} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i}$. See appendix A.5 for the full derivation, and [4] for a discussion about convergence of the differential entropy estimator to the true differential entropy value. The estimator for the expected entropy of $\hat{b}_t$, (2.14), is also a function of $\hat{b}_{t-1}$, hence the reward structure $R(\hat{b}, a, \hat{b}')$.

Similar to the discrete case, we use an abstract observation model (2.4), where instead of discrete observations, summation is done over observation samples, see Figure 2.2. We obtain upper and lower bounds that resemble the results (2.8), (2.9) and (2.12) but depend on the approximate expected reward value. Combining results on the expected state-dependent reward and expected entropy,

Theorem 2.3. *The estimated expected reward is bounded by,*

$$0 \leq \hat{\bar{\mathbb{E}}}_z \left[R\left(\hat{b}, a, \hat{\bar{b}}'\right)\right] - \hat{\mathbb{E}}_z \left[R\left(\hat{b}, a, \hat{b}'\right)\right] \leq \omega_2 log(K). \tag{2.15}$$

Proof. see appendix A.6. ■

Here, $\hat{\bar{b}}' \doteq \{\bar{q}^i, x^i\}_{i=1}^N$ denotes a particle set with abstract weight, $\bar{q}^i$, due to the abstract observation model (2.4), and

$$\hat{\bar{\mathbb{E}}}_z[\cdot] \doteq \sum_{m=1}^{M} \sum_{i=1}^{N} \bar{O}(z^m \mid x^i) q_{t-1}^i[\cdot]. \tag{2.16}$$

$\hat{\mathbb{E}}_z[\cdot]$ and $\hat{b}'$ are defined similarly by replacing the abstract model with the original one. As a result of Theorem 2.3, the estimated value function is bounded,

Corollary 2.4. *The difference between the estimated value function and the abstracted value function bounded by,*

$$0 \leq \hat{\bar{V}}^\pi(b_t) - \hat{V}^\pi(b_t) \leq \mathcal{T} \cdot \omega_2 log(K). \tag{2.17}$$

Proof. see appendix A.7. ■

The computational complexity of the expected reward in (2.15) is dominated by the complexity of the expected entropy, (2.14), which is $O(MN^2)$. By simply choosing $K = M$ the time complexity of the abstract expected reward diminishes to $O(N^2)$ with bounded loss. Utilizing our result directly induces a trade-off between computational speed and approximation loss of the value function. In the next section we derive an adaptive algorithm that gains computational efficiency *without any loss* in terms of the selected action.

2.3 Algorithms

Since the derivations in previous sections are agnostic to which algorithm is being used, we begin this section by presenting the contribution of our work to an existing algorithm. Then, based on insights gained from the examined algorithm, we propose modifications to improve the current algorithm. In the following section, we show that the changed algorithm empirically surpasses the current SOTA in performance throughout our experiments by a significant margin.

2.3.1 Baseline Algorithms

Sparse sampling (SS) algorithm, introduced in [24], provides ϵ-accuracy guarantee on the solution at a finite time. However, since it searches the tree exhaustively, the convergence is quite slow in practice. On the other hand, MCTS algorithm [6] has the desirable property of focusing its search on the more promising parts of the tree, but was shown to have poor finite-time performance, requiring an $exp(exp(...exp(1)...))$[1] iterations in the worst-case scenario [37]. To combat the worst-case running time of MCTS and the slow running time of SS, Forward Search Sparse Sampling (FSSS)

[1] A composition of D-1 exponentials, where D denotes tree depth.

Algorithm 2.1 AI-FSSS

Procedure: SIMULATE(b,d)

if d = 0 **then**
  Return 0, 0
else if $|C(b)| < |\mathcal{A}|$ **then**
  $ba, a, z_{\{1,\dots,K\}} \leftarrow$GEN$(b, K)$
  $\bar{P}_{z|x} \leftarrow$ ABSTRACTOBS$(ba, z_{\{1,\dots,K\}})$ // eq.(4)
  $\bar{\mathbb{E}}[\mathcal{R}(ba)] \leftarrow$EXPECTEDREWARD$(b, a, ba, \bar{P}_{z|x})$
else
  $a \leftarrow$ SELECTACTION(b)
end if
$lb \leftarrow \bar{\mathbb{E}}[\mathcal{R}(ba)]$
$ub \leftarrow \bar{\mathbb{E}}[\mathcal{R}(ba)] + log(K)$
if $0 < N(ba) < K$ **then**
  $z \leftarrow$ POP $(z_{\{1,\dots,K\}})$
  $b' \leftarrow$ POSTERIOR(b, a, z)
  $V_{LB}, V_{UB} \leftarrow$SIMULATE$(b', d-1)$
else if $N(ba) = K$ **then**
  $b' \leftarrow \arg\min_{b'} N(b')$
  $V_{LB}, V_{UB} \leftarrow$SIMULATE$(b', d-1)$
else if $N(ba) = 0$ **then**
  $V_{LB}, V_{UB} \leftarrow$ROLLOUT$(ba, d-1)$
end if
$LB(ba) \leftarrow lb + \frac{V_{LB} + (|C(ba)|-1)(LB(ba)-lb)}{|C(ba)|}$
$UB(ba) \leftarrow ub + \frac{V_{UB} + (|C(ba)|-1)(LB(ba)-ub)}{|C(ba)|}$
$a^* \leftarrow \arg\max_a UB(ba)$
$LB(b) \leftarrow LB(ba^*)$
$UB(b) \leftarrow UB(ba^*)$
$N(b) \leftarrow N(b) + 1$
$N(ba) \leftarrow N(ba) + 1$
Return $LB(b), UB(b)$

[55] was introduced. It was shown to achieve comparable performance to MCTS with performance guarantees under finite computational budget as in SS.

2.3.2 FSSS with Information-Theoretic Rewards

In its original version, FSSS introduced lower and upper bounds on the estimate of the $Q^d(x, a)$ function. In contrast to SS, FSSS builds the tree incrementally, where each iteration begins at the root node and proceeds down to horizon H, to obtain an estimate for the action-value function. Whenever a new node is expanded, its direct action-nodes are created alongside M randomly sampled children for each of them. The branching factor, M, is a predefined hyper-parameter. After performing $(|A| \cdot M)^d$ iterations, FSSS builds the same tree as SS, but enjoys anytime properties. Moreover, FSSS may benefit from reduced computation by utilizing upper and lower bounds and

Algorithm 2.2 SOLVE

Procedure: SOLVE

1: **for** $i \in 1 : n$ **do**
2: SIMULATE(b_{init}, d_{max})
3: **end for**
4: $action \leftarrow$ ADAPTBOUNDS(b_{init})
5: **Return** *action*

pruning actions that are sub-optimal.
When the reward is defined as an information-theoretic function, such as differential entropy, upper and lower bounds on the value function cannot be determined a-priori, thus no pruning can be made. Nonetheless, our experimental evaluations suggest that FSSS still serves as a strong baseline.

2.3.3 Adaptive Information-FSSS

We coin our new algorithm Adaptive Information FSSS (AI-FSSS). Given the same number of iterations, the actions obtained by the two algorithms are identical. The pseudo-code presented in appendix A.2. As in FSSS, we build the tree incrementally, where each iteration adds a new trajectory to the tree. The algorithm constructs an abstract belief tree, where a set of K posterior beliefs share the same reward upper and lower bounds relating it to the underlying reward value. This is done by immediately sampling K observation samples whenever a new action node expanded, followed by a computation of the abstract reward. Based on Theorem 2.3 we derive an ADAPTBOUNDS procedure, that adapts the number of aggregated observations. Bounds adaptation halts whenever the highest lower bound, $\max_a LB(b_{init}a)$, is higher than the upper bound of any other action. This results in the same action selection for the full FSSS and our adaptation, AI-FSSS.

2.3.4 Introducing Rollouts to AI-FSSS

A direct adaptation of FSSS to AI-FSSS would abstract K observations in each new action node up to the full depth of the tree, d_{max}. However, when the time budget is limited, it might not be the best strategy, since the abstraction of deeper belief nodes of the tree may never be visited twice, but might need to be refined afterward. Instead, we propose to perform rollout whenever a new action node is met for the first time. This is similar to MCTS, where rollouts are used to get an estimate of the action-value function. This approach will lead to abstraction only for expanded nodes, which are the ones in proximity to the root node. As the number of iterations grows, action nodes are expanded gradually and more abstract belief nodes are added to the tree. Given that the number of iterations equals the number of action nodes in the original Sprase-Sampling tree, followed by ADAPTBOUNDS procedure, both algorithms converge to the same solution.

Algorithm 2.3 REFINE

Procedure: REFINE(b,ba,d)

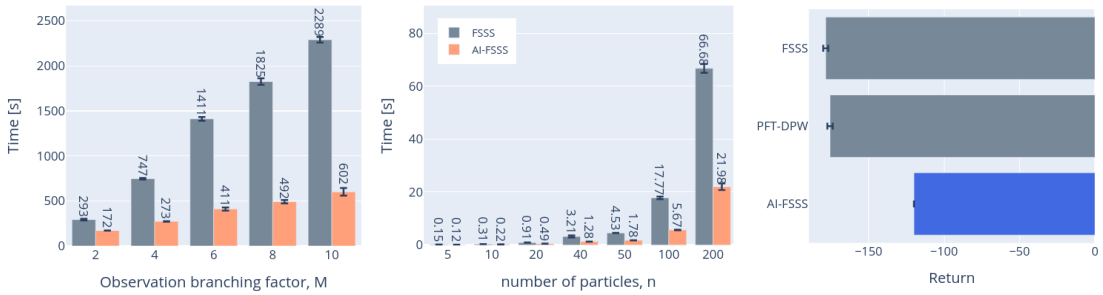
if IsLeaf(b) **then**
  Return 0, 0
else if ABSTRACT(ba) **then**
  $r_{old} \leftarrow$REUSEREWARD(ba)
  $P_{z|x} \leftarrow$ ORIGINALOBSMODEL($ba, z_{\{1,\dots,K\}}$)
  $\mathbb{E}[\mathcal{H}(ba)] \leftarrow$EXPECTEDENTROPY($b, ba, P_{z|x}$)
  $r \leftarrow r_{old} + \omega_2(\mathbb{E}[\mathcal{H}(ba)] - \bar{\mathbb{E}}[\mathcal{H}(ba)])$
else
  $r \leftarrow$REUSEREWARD(ba)
end if
$b' \leftarrow \arg\max_{b'}(UB(b') - LB(b'))$
$a' \leftarrow \arg\max_{a'}(UB(b'a') - LB(b'a'))$
$V_{LB}, V_{UB} \leftarrow$REFINE($b', b'a', d-1$)
$LB(ba) \leftarrow lb + \frac{V_{LB}+(|C(ba)|-1)(LB(ba)-lb)}{|C(ba)|}$
$UB(ba) \leftarrow ub + \frac{V_{UB}+(|C(ba)|-1)(LB(ba)-ub)}{|C(ba)|}$
$a^* \leftarrow \arg\max_a UB(ba)$
$LB(b) \leftarrow LB(ba*)$
$UB(b) \leftarrow UB(ba*)$
return $LB(b), UB(b)$

Procedure: ADAPTBOUNDS(b_{init})

while $\max_{a^+ \in \mathcal{A}} LB(b_{init}a^+) < \max_{a \in \mathcal{A} \setminus a^+} UB(b_{init}a)$ **do**
  $a^* \leftarrow \arg\max_{a \in \mathcal{A}} LB(b_{init}a)$
  REFINE($b_{init}, b_{init}a^*, d$)
end while
return a^*

2.3.5 Implementation

In this section we present the main building blocks to derive our algorithm. The variables used in Algorithm A.1 are b, ba, and b' which represent a belief node, a predicted belief node, i.e. after performing an action and posterior belief, after incorporating a measurement. $C(\cdot)$ denotes a list of their corresponding children. a and $z_{\{1,\dots,K\}}$ denote an action and a list of K sampled observations respectively. $\bar{P}_{z|x}$ is a list holding the abstract probability values of the measurement model, as in equation (4). $R_{state}(\cdot,\cdot)$ denotes a state-dependent reward function, which may be defined arbitrarily. LB, UB and N are all initialized to zero. ROLLOUT performs a predefined policy. In our experiments, we chose uniform distribution over all actions for the rollout policy. Algorithm A.2 uses b_{init}, which represents the initial belief at the root node, n is the number of iterations and d_{max}, the maximum depth of the planning tree.



(a) (b) (c)

Figure 2.3: Evaluating performance of AI-FSSS. (a-b): Running time comparison of FSSS, and our adaptation, AI-FSSS without rollouts. Both algorithms end with the same action selection, but with increasing difference in computation time. (a) Different observation branching factor, with 20 particles. (b) Different number of state particles, with 4 observation after each action node. (c) Average total return of AI-FSSS with rollouts in 2D Light-Dark with obstacles.

2.4 Experiments

The goal of the experiment section is to evaluate the influence of the abstraction mechanism on the planning performance. We examined both the time difference and the total return. All algorithms use a particle filter for inference, the choice of the particle filter variant is independent of our contribution. All experiments were performed on the common 2D Light Dark benchmark, where both the state and observation spaces are continuous; see an illustration in Figure 2.4. In this problem, the agent is required to reach the goal while reducing localization uncertainty using beacons scattered across the map. The reward function defined as a weighted sum of distance to goal, which is state dependent reward and entropy, as in (2.1). Due to space limitations, domain and implementation details are deferred to appendix A.2.1.

2.4.1 Time Performance Evaluation

We compared the basic FSSS with our adaptation, AI-FSSS, in terms of time efficiency. As stressed in previous sections, both algorithms guaranteed to select the same action. To ensure that both algorithms built the same tree, rollouts were avoided and each iteration proceeded until the maximum depth of the tree. We note that the expected return was inferior to our full algorithm, which is evaluated next. Technically, we also fixed the random numbers by selecting the same seed in both algorithms.

Observation branching factor. In the first experiment, we fixed the number of state particles to $n = 40$, and examined the influence of different branching factors over the observation space, see Figure 2.3. The algorithms were limited to 20,000 iterations before performing an action. The computation time indicates an empirical

average of over 1,000 simulations, approximating the mean time for a full trajectory. As one would expect, the more observations are clustered in each aggregate, the more time-efficient AI-FSSS compared to the basic FSSS. To obtain the *same* best action in both algorithms, the tree construction was followed by a refinement step, see appendix A.2, AdaptBounds. In the adapt-bounds step, we incrementally reduce the number of aggregates, so in the worst-case every aggregate will only hold a single observation, and thus recover the FSSS tree. This will occur only in a degenerate case, where all action-values $Q(b_0, \cdot)$ will have the same value, which is rarely the case.

Number of particles. In our second experiment, we evaluated the effect of the number of particles representing the belief. Here, the number of observations was fixed to $M = 4$. Figure 2.3 shows the change in computational speed with regard to the number of particles in our experiments. Both algorithms performed 1,000 simulations; The empirical running time mean and standard deviation are presented in the graph. In the experiments where only few particles were used, e.g. 5, the efficiency gain was mild. In a setting where few particles are sufficient, computing the entropy is relatively cheap compared other parts of the algorithm, which become relatively more significant (e.g. the different max operators). However, we observed that the burden became significant even in a mild number of particles, e.g. when $n = 20$ the speed-up ratio more than doubled while only a modest cluster size of 4 observations was used.

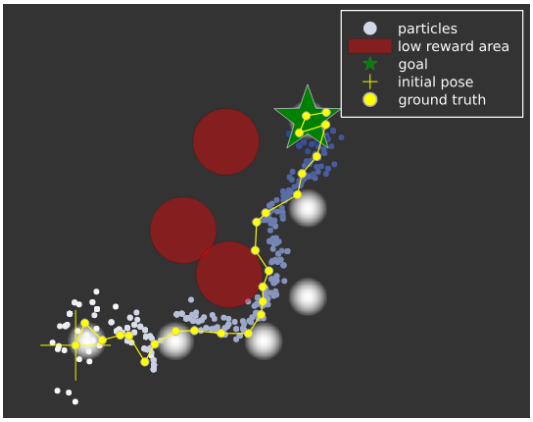


Figure 2.4: An illustration of the environment being used in our experiments.

2.4.2 Total Return Evaluation

In contrast to the previous experiments, in this section we evaluate the full version of AI-FSSS, that is, with rollouts for every newly expanded action node. We evaluated performance against FSSS [55] and PFT-DPW [48], by augmenting a POMDP to a belief-MDP. In this setting, each node holds n particles. All algorithms had 1 second limitation for planning before each interaction with the environment. Except for PFT-

DPW, the algorithms shared the same observation branching factor, $M = 4$. PFT-DPW opens new observation nodes as time progresses, depending on hyper-parameters defined by the expert. We identified k_o as the dominant hyper-parameter that controls the number of observations in our experiment. We experimented with both $k_o = 4$ and $k_o = 2$ in order to keep a comparable observation branching size. The better result is shown here, see Figure 2.3.

Each algorithm performed 1,000 full trajectories in the environment, each contained 25 steps. As suggested in Figure 2.3, our algorithm performed better than PFT-DPW and FSSS when information-gathering was an explicit part of the task. The superior results are expected due to the additional efficiency of our approach. Both FSSS and PFT-DPW compute the expensive-to-evaluate reward value for every newly expanded node, whereas AI-FSSS compute the exact reward only when it is required in order to determine the best action selection. Consequently, under a given time-limit, AI-FSSS expands more posterior nodes in the belief tree, which result in better coverage of the belief tree.

2.5 Conclusions

This chapter deals with online planning under uncertainty with information-theoretic reward functions. Information-theoretic rewards facilitate explicit reasoning about state uncertainty, contrary to the more common expected reward over the state. Due to the added computational burden of evaluating such measures, we consider an observation model abstraction that improves efficiency. We derived analytical bounds with respect to the original reward function. Additionally, we introduced a new algorithm, AI-FSSS, that contracts the bounds upon need, and is guaranteed to select identical action as the vanilla algorithm. Finally, we conducted an empirical performance study with and without observation abstraction. Our results suggest a significant speed-up as the cardinality of the particle set and the observation-branching factor increases while yielding same performance.

Chapter 3

Monte Carlo Planning in Hybrid Belief POMDPs

While in the previous chapter we have simplified the reward calculation by considering a simplified observation space, in this chapter we generalize this idea by simplifying the entire tree construction and the value function using simplified state space.

More specifically, we consider a POMDP setting with hybrid state space that results in a belief containing discrete and continuous random variables. While the states of the agent and of the environment are commonly represented by continuous random variables, discrete random variables generally represent object classes, data association hypotheses or even transition models (e.g. due to slippage) and observation models. In ambiguous environments, where different objects or scenes can possibly be perceptually similar or identical, such discrete variables are particularly important, as wrong assignments can lead to a complete failure of the agent's task.

In this chapter we propose an approach to alleviate the computational complexity of planning with hybrid beliefs under the POMDP formulation. We show that previous algorithms result in biased estimators of the reward and value function, and suggest a different way for controlling the number of hypotheses to a manageable size. Utilizing sequential importance resampling (SIR) for hypothesis selection, we suggest an algorithm that results in an unbiased estimator and efficient belief tree construction. We show that the algorithm supports both state-dependent and belief-dependent rewards. We proceed with a contribution to inference in the setting of ambiguous data association, by introducing a natural way to incorporate negative information within Bayesian inference, and demonstrate how the hypotheses weights should be updated. Last, we demonstrate our approach on simulative environments to corroborate our findings.

Our contributions are as follows: (a) We introduce a novel algorithm that performs Monte-Carlo planning to solve a POMDP when the considered belief is hybrid. (b) We show that our algorithm, HB-MCP, leads to an unbiased utility estimate, in contrast to existing hybrid belief algorithms. (c) We introduce negative information to hybrid belief inference. (d) We demonstrate the effectiveness of our algorithm in

extremely aliased simulated environments where unresolved data association leads to multi-modal belief hypotheses. To maintain fluency of reading, the formal proofs and further implementation and experimental details are located in appendix A.3.

3.1 Preliminaries

Based on the POMDP definition as described in section 1.2.1, we introduce the following chapter-specific additions and changes to the definition; Given a finite planning horizon $\mathcal{T}$ the value function for a policy π is defined as the expected cumulative reward received by executing π,

$$V^{\pi}(b_t) = R(b_t, \pi(b_t)) + \mathop{\mathbb{E}}_{z_{t+1:\mathcal{T}}} \left[\sum_{\tau=t+1}^{\mathcal{T}} R(b_\tau, \pi(b_\tau)) \right]. \tag{3.1}$$

Similarly, an action-value function,

$$Q^{\pi}(b_t, a_t) = R(b_t, a_t) + \mathop{\mathbb{E}}_{z_{t+1}} \left[V^{\pi}(b_{t+1}) \right], \tag{3.2}$$

is defined by executing action a_t and then following the policy π for a finite horizon $\mathcal{T}$. At each planning session, the agent solves a POMDP by searching for the optimal policy π^* that maximizes (3.1). Note that $R(b, a)$ is a general reward function on the belief and action. In the following chapter, we discern between reward functions that are restricted to state dependent functions or general belief dependent functions, and use the notations $\mathcal{R}_X \triangleq \mathbb{E}_{X\sim b}[r(X, a)]$ and $\mathcal{R}_b$ as a shorthand to make it clear which definition is being considered. Here, X denotes a generalized state which includes the current time step, and past time steps, more concretely, $X_t \triangleq \{x_0, .., x_t\}$.

3.1.1 Hybrid Belief

A hybrid belief is defined over both continuous and discrete random variables. The continuous random variables can represent the state of the agent and (possibly also) of the environment, as common in SLAM framework. The discrete random variables can represent, e.g., object classes and/or data association hypotheses. Nevertheless, the following definition is general and not restricted to these examples.

We formally define the hybrid belief at each time t as

$$b_t \triangleq \mathbb{P}(X_t, \beta_{0:t} \mid H_t) = \underbrace{\mathbb{P}(X_t \mid \beta_{0:t}, H_t)}_{b[X_t]_{\beta_{0:t}}} \underbrace{\mathbb{P}(\beta_{0:t} \mid H_t)}_{b[\beta_{0:t}] \equiv \omega_t}, \tag{3.3}$$

where $H_t \triangleq \{z_{1:t}, a_{0:t-1}\}$ represents all past actions and observations. $b[X_t]_{\beta_{0:t}}$ is the conditional belief over continuous variables. ω_t is the marginal belief over discrete variables which can be considered as the hypothesis weight. We define $H^-_{t+1} \triangleq H_t \cup \{a_t\}$ and $b^-_{t+1} \triangleq \mathbb{P}\left(X_{t+1}, \beta_{0:t+1} | H^-_{t+1}\right)$ for notational convenience.

The marginal belief ω_t is updated for each realization of discrete random variables according to

$$\begin{aligned}\omega_t^{i,j} &= \eta^{-1}\mathbb{P}(z_t \mid \beta_{0:t}^{i,j}, H_t^-)\mathbb{P}(\beta_{0:t}^{i,j} \mid H_t^-) \\ &= \eta^{-1}\overbrace{\mathbb{P}(z_t \mid \beta_{0:t}^{i,j}, H_t^-)\mathbb{P}(\beta_t^i \mid \beta_{0:t-1}^j, H_t^-)}^{\zeta_t^{i|j}}\overbrace{\mathbb{P}(\beta_{0:t-1}^j \mid H_t^-)}^{\omega_{t-1}^j},\end{aligned} \tag{3.4}$$

which is obtained by Bayes rule followed by chain rule on ω_t. The un-normalized weight can be expressed recursively as $\tilde{\omega}_t^{i,j} = \zeta_t^{i|j}\omega_{t-1}^j$. The conditional belief $b[X_t]_{\beta_{0:t}}$ is updated for each realization of discrete random variables as

$$b[X_t]_{\beta_{0:t}}^{i,j} = \psi(b[X_t]_{\beta_{0:t-1}}^j, a_{t-1}, z_t), \tag{3.5}$$

where $\psi(.)$ represents the Bayesian inference method.

Generally, when planning with hybrid beliefs the agent constructs both a belief tree and multiple hypotheses trees. Each hypotheses tree represent the posterior hypotheses given a history. Since every node of the planning tree (i.e. belief tree) corresponds to a hypotheses tree, the computational complexity of the corresponding POMDP becomes a significant burden. In the following section we present a novel algorithm that circumvent this difficulty via Monte-Carlo sampling.

3.2 POMDP Planning with Hybrid Beliefs

This section starts with a brief overview of how MCTS can be utilized to solve POMDPs with hybrid beliefs and its drawbacks. Then, we present a novel approach to utilize the UCT exploration bonus to build an asymmetric hypotheses tree, which leads to better use of the computational resources by optimistically focusing on the interesting hypotheses.

3.2.1 vanilla Hybrid-Belief MCTS

For completeness, we first present a vanilla-HB-MCTS algorithm. Although the exact algorithm does not seem to exist in the literature, this is the ad-hoc way to interleave hybrid beliefs with state-of-the-art POMDP solvers. vanilla-HB-MCTS, can be seen as an adaptation of the state-dependent MCTS [45] algorithm to a (hybrid-)belief (3.3), by augmenting the belief to a belief-state. A similar approach was also taken by PFT-DPW [48], which utilized particle filters to approximate a posterior belief, over continuous variables. However, computing a full hybrid belief is a difficult and sometimes intractable task, even for particle-based solvers, and is thus prone to approximations.

Pruning. The number of hypotheses at each posterior node in the belief tree may be prohibitively large. To handle the infeasible number of the posterior hypotheses,

vanilla-HB-MCTS utilizes a pruning mechanism similar to those suggested in [40, 18]. As a result, unlikely hypotheses are removed from the hypotheses tree.

In vanilla-HB-MCTS, each posterior node holds a fixed number of hypotheses once expanded, depending on a predefined hyperparameter. Such a method may sometimes be too harsh, pruning away hypotheses with high probability due to a limited hypotheses budget, or too loose, keeping highly unlikely hypotheses, thus wasting valuable computational time. Other approaches may also be applicable, such as fixing a probability threshold value, under which all hypotheses are pruned. However, the latter has its own deficiencies, such as hypothesis depletion. For completeness, we describe vanilla-HB-MCTS implementation details in the appendix A.3.

3.2.2 Hybrid Belief Monte-Carlo Planning

In contrast to vanilla-HB-MCTS, in HB-MCP, we do not use any pruning heuristic for two reasons: (1) this requires knowledge, or an insight, as to how many hypotheses would be sufficient for the specific POMDP; (2) Each posterior node in the belief tree maintains hypotheses based on a hyperparameter, regardless of how relevant this node may be for decision-making.

Conversely, we suggest an adaptive algorithm that focuses computational resources in proportion to their relevance in the belief tree, which circumvent the difficulty in full belief update. HB-MCP is recursively invoked with a single sampled hypothesis. Every such single hypothesis may evolve into multiple hypotheses. HB-MCP algorithm computes only the posterior weights (i.e. probability values) that are conditioned on that single hypothesis, followed by a random weight sample based on their categorical distribution. Then, only the hypothesis associated with the sampled weight is updated. This is in contrast to the full posterior update done in vanilla-HB-MCTS.

Additionally, to support belief-dependent rewards, the reward value is estimated based on state samples received across multiple visits to the belief node, i.e., state samples from multiple hypotheses. We describe the algorithm details in section 3.3.

HB-MCP holds some desirable properties compared to the full belief update and pruning approaches. First, at each iteration of HB-MCP, a maximum of $\mathcal{T}$ posterior hypotheses are computed, and a small subset of the weights. This is in contrast to the full posterior update, that would require the entire (or pruned-)set of the current posteriors, and compute all the posterior hypotheses of the next time-step, which is highly resource expensive for every iteration. Second, HB-MCP explores both the planning tree and the hypotheses trees by focusing its computational effort on the interesting parts, utilizing UCB to guide the search; this property is inspired by MCTS which builds the planning tree by focusing on the optimistic parts of the tree. In section 3.4, we show that this approach results in an unbiased estimator for the true value function.

Algorithm 3.1 HB-MCP

Procedure:SIMULATE(b_t^j, h, d)
1: **if** d = 0 **then**
2: **Return** 0
3: **end if**
4: $a \leftarrow \arg\max_{\bar{a}} Q(h\bar{a}) + c\sqrt{\frac{log(N(h))}{N(h\bar{a})}}$
5: $B(h) \leftarrow$GETSAMPLES($b_t^j, B(h), N(h)$)
6: $r \leftarrow$ REWARD($B(h), a$)
7: $r \leftarrow r + N(h)(r - r_{prev})$
8: **if** $|C(ha)| \leq k_o N(ha)^{\alpha_o}$ **then**
9: $z \leftarrow$ SAMPLEOBSERVATION(b_t^j, a)
10: **else**
11: $z \leftarrow$ Sample uniformly from $C(ha)$
12: **end if**
13: $\{\omega_{t+1}^{i,j}\}_{i=1}^{L} \leftarrow$ COMPUTEWEIGHTS(b_t^j, a, z)
14: $i \leftarrow$ SAMPLECATEGORICAL($\{\omega_{t+1}^{i,j}\}_{i=1}^{L}$)
15: $b_{t+1}^{i,j} \leftarrow \Psi(b_t^j, a, z, i)$ // *Eq.* (3.5)
16: **if** $z \notin C(ha)$ **then**
17: $C(ha) \cup \{z\}$
18: $R \leftarrow r +$ROLLOUT($b_{t+1}^{i,j}, d-1$)
19: **else**
20: $R \leftarrow r +$SIMULATE($b_{t+1}^{i,j}, haz, d-1$)
21: **end if**
22: $N(h) \leftarrow N(h) + 1$
23: $N(ha) \leftarrow N(ha) + 1$
24: $Q(ha) \leftarrow Q(ha) + \frac{R - Q(ha)}{N(ha)}$
25: **Return** R

3.3 Implementation details

In this section we describe the implementation details of our approach, HB-MCP, as discussed in section 3.2.2.

HB-MCP can be described as follows; first, it starts by receiving a single hypothesis and selecting an immediate action according to UCB exploration bonus. Then, samples are generated and appended to $B(h)$, which are later used for reward estimation (lines 5- 7). Lines 8-11 perform observation progressive widening. Then, the approach for sampling hypotheses is shown in lines 13-15. Note that the algorithm directly computes *all* the weights conditioned on the hypothesis given as input (line 13). Then, we re-sample a *single* conditional belief, $b_{t+1}^{i,j}$, sampled according to the weights (line 14). We note that this is not a necessity, and different number of samples can be taken in those two steps to trade-off efficiency and accuracy. Depending on whether a new posterior node is sampled or not, lines 16-20 either call for rollout or continues recursively. Last, the action-value function and the counters are updated.

To estimate a belief-dependent reward, state samples should correspond to their likelihood in the full hybrid belief. In HB-MCP, hypotheses are generated iteratively, accumulating hypotheses (or, equivalently, state samples from those hypotheses), so that at each iteration the reward estimator is improved. Generally, a belief dependent reward is not a simple average over samples. However, as in MCTS, HB-MCP estimates the action value function, $Q(ha)$, as an average of all the cumulative returns passed through that node. To support belief dependent rewards, HB-MCP computes a new

reward estimate based on all past samples, and replaces the previous reward estimate with the new one. To that end, a simple recursive subtraction and addition update is done for every node encountered along the path of the current iteration, described in line 7.

3.4 Theoretical Analysis

In this section, we first claim that existing approximations, done in contemporary state-of-the-art multi-hypotheses planners, such as DA-BSP [40], ARAS [19] as well as vanilla-HB-MCTS (Section 3.2.1), lead to a biased estimation of the reward value, and therefore a biased value function. Further, we show that even if the reward value could be precisely recovered, the resultant value function is generally biased. Instead, HB-MCP performs sequential sampling which converges to the correct value. Then, we discuss how HB-MCP may also support belief-dependent reward functions and its applicability for value function estimation.

3.4.1 State-dependent rewards

State-dependent reward functions are defined as the expected reward value over the belief, i.e., $\mathcal{R}_X \triangleq \mathbb{E}_{X\sim b}[r(X,a)]$. Generally, state-dependent rewards cannot be computed analytically, thus, they are approximated using state samples. Since in a hybrid belief the number of hypotheses may be prohibitively expensive to compute, most existing algorithms approximate the belief, $\hat{b}$, by performing some heuristic pruning. As a consequence, the approximate distribution is shifted, and the reward value is biased even with an infinite number of state samples,

Lemma 3.4.1. *The estimator $\mathbb{E}_{X\sim\hat{b}}[r(X,a)]$ is biased.*

Proof. Assuming the weights of the pruned hypotheses are non-zero, the proof is immediate,

$$\begin{aligned}\mathbb{E}_{X\sim b}[r(X,a)] &= \int_X \sum_\beta b(X,\beta)r(X,a)dX \qquad (3.6)\\ &= \int_X \sum_{\beta\in A} b(X,\beta)r(X,a)dX + \sum_{\beta\in\neg A} b(X,\beta)r(X,a)dX\\ &\neq \eta_A \int_X \sum_{\beta\in A} b(X,\beta)r(X,a)dX = \mathbb{E}_{X\sim\hat{b}}[r(X,a)].\end{aligned}$$

where A denotes the set of un-pruned hypotheses, and η_A is their corresponding normalizer after pruning. ■ ■

In contrast, HB-MCP samples hypotheses iteratively starting from the root node; it utilizes sequential importance resampling, which results in an unbiased estimator for the reward value. At every iteration, the new sampled states from the current hypothesis

are added to the estimator from previous iterations, by averaging. The process for generating hypotheses can be described as follows; for any time t, a hypothesis is sampled i.i.d from a proposal-prior distribution, $\beta_0^i \sim \mathbb{Q}(\beta_0 \mid H_0)$. Then, hypotheses are recursively sampled from a proposal distribution, $\beta_\tau^i \sim \mathbb{Q}(\beta_\tau \mid \beta_{0:\tau-1})$ up to time $\tau = t$. We define $\mathbb{Q}(\beta_0 \mid H_0) \triangleq \mathbb{P}(\beta_0 \mid H_0)$, and $\mathbb{Q}(\beta_\tau \mid \beta_{0:\tau-1}) \triangleq \textsc{Uniform}\,[1, |\beta_\tau|]$. Then, for every time-step t, the corresponding importance weight is,

$$\begin{aligned}\lambda_t^{i,j} &= \frac{\mathbb{P}(\beta_{0:t}^{i,j} \mid H_t)}{\mathbb{Q}(\beta_{0:t}^{i,j} \mid H_0)} = \frac{\eta_t \zeta_t^{i|j} \mathbb{P}(\beta_{0:t-1}^j \mid H_{t-1})}{\mathbb{Q}(\beta_t^i \mid \beta_{0:t-1}^j)\mathbb{Q}(\beta_{0:t-1}^j \mid H_0)} \\ &= \frac{\eta_t \zeta_t^{i|j}}{1/|\beta_t^{i|j}|} \frac{\mathbb{P}(\beta_{0:t-1}^j \mid H_{t-1})}{\mathbb{Q}(\beta_{0:t-1}^j \mid H_0)} = \eta_t \zeta_t^{i|j} |\beta_t^{i|j}| \lambda_{t-1}^j,\end{aligned} \tag{3.7}$$

where $\lambda_0^j = 1$. As a consequence,

Lemma 3.4.2. *HB-MCP state-dependent reward estimator,* $\hat{\mathcal{R}}_X \triangleq \frac{1}{N}\sum_{i,j=1}^N \lambda_t^{i,j} \frac{1}{n_X} \sum_{k=1}^{n_X} r(X_t^{i,j,k}, a_t)$, *is unbiased.*

Proof.. If states are sampled i.i.d. for each hypothesis, then the expected value of the reward estimator, $\hat{\mathcal{R}}_X$, is,

$$\begin{aligned}\mathbb{E}\left[\hat{\mathcal{R}}_X\right] &\triangleq \mathbb{E}\left[\frac{1}{N}\sum_{i,j=1}^N \lambda_t^{i,j} \frac{1}{n_X}\sum_{k=1}^{n_X} r(X_t^{i,j,k}, a_t)\right] \\ &= \mathbb{E}_{\mathbb{Q}}\left[\frac{1}{N}\sum_{i,j=1}^N \lambda_t^{i,j} \mathbb{E}_{b[X_t]_{\beta_{0:t}}^{i,j}}\left[\frac{1}{n_X}\sum_{k=1}^{n_X} r(X_t^{i,j,k}, a_t)\right]\right] \\ &= \frac{1}{N}\sum_{i,j=1}^N \mathbb{E}_{\mathbb{Q}}\left[\frac{\mathbb{P}}{\mathbb{Q}}\frac{1}{n_X}\sum_{k=1}^{n_X} \mathbb{E}_{b[X_t]_{\beta_{0:t}}}\left[r(X_t^{i,j,k}, a_t)\right]\right] \\ &= \mathbb{E}_{\mathbb{P}}\left[\mathbb{E}_{b[X_t]_{\beta_{0:t}}} r(X_t, a_t)\right] \triangleq \mathcal{R}_X\end{aligned} \tag{3.8}$$

where $\mathbb{P} = \mathbb{P}(\beta_{0:t} \mid H_t)$, $\mathbb{Q} = \mathbb{Q}(\beta_{0:t} \mid H_t)$, and N and n_X denote the number of samples from $\mathbb{Q}$ and $b[X_t]_{\beta_{0:t}}^{i,j}$ respectively. ■ ■

As the planning horizon grows, sampling hypotheses uniformly quickly induce sample degeneracy. That is, the weights of most hypothesis samples become negligible, while only a few remain significant, which negatively affects the accuracy of the estimate. To avoid this issue, we perform resampling at every step, also known as sequential importance resampling (SIR). Before resampling, each hypothesis weight simply becomes, $\lambda_t^{i|j} = \eta_t \zeta_t^{i|j} \left|\beta_t^{i|j}\right|$, which is then updated to $1/N$ after resampling. Note that resampling does not introduce bias to the estimator [25]. To avoid repeated derivations, for the rest of this sequel we treat mathematical proofs as if hypotheses are directly sampled from distribution $\mathbb{P}$, even though they are in fact sampled from the proposal distribution, $\mathbb{Q}$. However, all derivations can be started by sampling from $\mathbb{Q}$, then follow similar steps of lemma A.3.1 followed by resampling to arrive at the same result.

In some cases of interest, such as ambiguous DA, the normalizer η_t cannot be easily computed, and so the importance weight, λ_t, cannot be computed. A common practice is to use the self-normalized version of the estimator, i.e. $\tilde{\lambda}_t^{i|j} = \tilde{\lambda}_{t-1}^{i|j} \frac{\zeta_t^{i|j}}{\sum \zeta_t^{i|j}}$, which is no longer unbiased [25]. However, the self-normalizing variation is consistent, meaning it becomes less biased with more samples and converges in probability (denoted $\to^p$) to the theoretical value. This is a direct consequence of applying the weak law of large numbers on both the nominator and denominator of the self-normalized estimator,

$$\begin{aligned} \hat{\mathcal{R}}_X^{SN} &\triangleq \frac{\sum_{i,j=1}^{N} \zeta_t^{i|j} \omega_{t-1}^{j} \frac{1}{n_X} \sum_{k=1}^{n_X} r(X_t^{i,j,k}, a_t)}{\sum_{i,j=1}^{N} \zeta_t^{i|j} \omega_{t-1}^{j}} \\ &= \frac{\frac{1}{N} \sum_{i,j=1}^{N} \eta_t \zeta_t^{i|j} \omega_{t-1}^{j} \frac{1}{n_X} \sum_{k=1}^{n_X} r(X_t^{i,j,k}, a_t)}{\frac{1}{N} \sum_{i,j=1}^{N} \eta_t \zeta_t^{i|j} \omega_{t-1}^{j}} \to^p \frac{\mathcal{R}_X}{1}, \end{aligned} \tag{3.9}$$

where the denominator converges to the sum of weights, $\sum_{i,j} \omega_t^{i,j} = 1$ and the nominator to the reward value.

3.4.2 Belief-dependent rewards

Contrary to state-dependent rewards, belief dependent rewards are not necessarily linear in the belief, so averaging over state samples from different hypotheses does not guarantee convergence to the theoretical reward value. Moreover, different reward definitions may be functions of not only the states, but also the weights, the conditional beliefs, or the probability density values of the complete theoretical belief (such as Shannon's entropy [43] or differential entropy [2]). To support the various cases, we split our discussion into the parametric case, where the reward can be precisely calculated given a set of parametric conditional beliefs and the corresponding weights, and the nonparametric case, where the reward is estimated based on state and hypothesis samples.

HB-MCP supports belief-dependent rewards by accumulating conditional beliefs across multiple visitations of the same history (i.e. same node in the belief tree). The estimated weight of each conditional belief is the sample frequency of the corresponding hypothesis. That is, $\hat{\mathbb{P}}(\beta_{0:t}^{i,j} \mid H_t) \triangleq \hat{\omega}_t^{i,j} = \frac{\sum_{i,j} \mathbf{1}_{\beta=\beta_{0:t}^{i,j}}}{N}$, where N is the number of hypothesis samples, $i, j \in [1, |\beta_{0:t}|]$, $|\beta_{0:t}|$ is the theoretical number of hypotheses at time t and $\mathbf{1}_{\square}$ denotes the indicator function.

Parametric. Assuming a parametric representation for the conditional beliefs, $b[X_t]_{\beta_{0:t}}^{i,j}$, the belief-dependent reward, $\mathcal{R}_b(b_t, a_t)$, is evaluated using the estimated hybrid belief, $\mathcal{R}_b(\hat{b}_t, a_t)$, where $\hat{b}_t = b[X_t]_{\beta_{0:t}} \hat{b}[\beta_{0:t}] \equiv b[X_t]_{\beta_{0:t}} \hat{\mathbb{P}}(\beta_{0:t} \mid H_t)$, and b_t defined in (3.3). Applying the hypothesis resampling approach as described in Section 3.4.1, the sample frequency of each hypothesis in $\hat{b}_t$ is unbiased, in other words, in expectation it equals the theoretical weights. Moreover,

Lemma 3.4.3. *$\mathcal{R}_b(\hat{b}_t, a_t)$ converges in probability to $\mathcal{R}_b(b_t, a_t)$ for any continuous, real-valued function $\mathcal{R}_b$.*

Proof.. By the law of large numbers, $\hat{\omega}_t^{i,j}$ is consistent as $N \to \infty$ for all $i, j \in [1, |\beta_{0:t}|]$,

$$\hat{\omega}_t^{i,j} = \sum_{k=1}^{N} \frac{\mathbf{1}_{\beta^k = \beta_{0:t}^{i,j}}}{N} \to^p \mathbb{P}(\beta_{0:t}^{i,j} \mid H_t) = \omega_t^{i,j}, \tag{3.10}$$

then, due to the continuous mapping theorem,

$$\mathcal{R}_b(b[X_t]_{\beta_{0:t}} \hat{b}[\beta_{0:t}], a_t) \to^p \mathcal{R}_b(b[X_t]_{\beta_{0:t}} b[\beta_{0:t}], a_t),$$

that is, $\mathcal{R}_b(\hat{b}_t, a_t)$ is a consistent estimator for $\mathcal{R}_b(b_t, a_t)$. ■ ■

Nonparametric. In the nonparametric case, the reward value is estimated based on state particles, which may correspond to conditional belief estimation via particle filters, or POMDPs with reward functions that have no close-form solution, and are thus approximated via Monte Carlo methods. Then, instead of $\mathcal{R}_b(b_t, a_t)$, an estimator over the reward is used, $\hat{\mathcal{R}}_b(\hat{b}[X_t]_{\beta_{0:t}} \hat{b}[\beta_{0:t}], a_t)$, where both the belief and the reward functions are estimators. We denote $\hat{b}[X_t]^k_{\beta_{0:t}} = \sum_{i=1}^{n_x} \alpha_t^{i,k} \delta(X - X_t^{i,k})$, where $\alpha_t^{i,k}$ is the weight of state particle i generated from conditional belief k and n_x is the number of particles used to approximate the conditional belief. To arrive at consistency results for an arbitrary nonparametric reward estimator, we assume that the reward estimator based on samples from the full theoretical belief is consistent, i.e., $\hat{\mathcal{R}}_b(\hat{b}[X_t]_{\beta_{0:t}} b[\beta_{0:t}], a_t) \to^p \mathcal{R}_b(b_t, a_t)$.

Lemma 3.4.4. *If $\hat{\mathcal{R}}_b(\hat{b}[X_t]_{\beta_{0:t}} b[\beta_{0:t}], a_t) \to^p \mathcal{R}_b(b_t, a_t)$, then $\hat{\mathcal{R}}_b(b[X_t]_{\beta_{0:t}} \hat{b}[\beta_{0:t}], a_t) \to^p \mathcal{R}_b(b_t, a_t)$.*

Proof.. The proof follows similar steps to lemma 3.4.3. ■ ■

3.4.3 Value function

When using the existing hypotheses pruning approximations, the estimated value function converges to the wrong value even when some external source provides the exact reward value. This is due to the way observations are generated. The value function is defined as

$$V^\pi(b_t) = \int_z \mathbb{P}(z_{t+1:\tau} \mid H_t^-) \sum_{\tau=t}^{\mathcal{T}} \mathcal{R}(b_\tau, \pi_\tau) dz, \tag{3.11}$$

and since there is usually no direct access to observations given history, first state-samples are generated, then observations are sampled using the observation model, that is, $\mathbb{P}(z_t \mid H_t^-) = \sum_\beta \int_X \mathbb{P}(z_t \mid X_t, \beta_{0:t}) b^-(X_t, \beta_{0:t})$. Replacing b^- with its pruned counterpart, $\hat{b}^-$, results in a shifted distribution for both the belief and the measurements, which impacts the value function estimation. Proof of this claim is similar to that of lemma 3.4.1 and skipped here for conciseness.

$z^{\beta_{t,k}}=\infty$	$\beta_{t,k}>n_{z_t}$	$(x^r,l^k)\in S.R.$	$\mathbb{P}(z \mid x,l)$	$\mathbb{P}(\beta \mid x,l)$
no	no	yes	$f(\cdot)$	1
no	no	no	0	0
yes	yes	no	1	1
yes	yes	yes	0	0
no	yes	yes	$f(\cdot)$	0
no	yes	no	0	1
yes	no	no	1	0
yes	no	yes	0	1

Table 3.1: Possible combinations when considering negative information. $z^{\beta_{t,k}}=\infty$ indicates no observation. Hypothesis element $\beta_{t,k} > n_{z_t}$ assumes that x^r, l^k are out of the sensing range. $(x^r,l^k) \in S.R.$ indicates that a specific realization is within the sensing range. $\mathbb{P}(z^{\beta_{t,k}} \mid x^r,l^k)$ and $\mathbb{P}(\beta_{t,k} \mid x^r,l^k)$ indicate the likelihood of the models. Last, $f(\cdot)$ denotes the likelihood value of the observation sensor (e.g. Gaussian).

Instead, HB-MCP generates observations by first receiving a hypothesis from the belief at the current node, $\beta^j_{0:t}$. Conditioned on $\beta^j_{0:t}$ and the history, HB-MCP samples a new plausible hypothesis, β^i_{t+1}. Then, an observation is sampled based on the posterior hypothesis. More formally,

$$\begin{aligned}\mathbb{E}_{z_{t+1:\tau}}[\sum_{\tau=t+1}^{\mathcal{T}}\mathcal{R}_\tau] &= \mathbb{E}_{z_{t+1}}\left[\mathcal{R}_{t+1}+\mathbb{E}_{z_{t+2:\tau}}\left[V^\pi_{t+2}\right]\right] \\ &= \underbrace{\mathbb{E}_{\beta_{0:t}}\mathbb{E}_{\beta_{t+1}|\beta_{0:t}}\mathbb{E}_{z_{t+1}|\beta_{0:t+1}}\left[\mathcal{R}_{t+1}\right]}_{\triangleq\alpha_{t+1}}+\mathbb{E}\left[V^\pi_{t+2}\right].\end{aligned} \tag{3.12}$$

We then define the estimator for the expected reward, $\hat{\alpha}_{t+1}$,

$$\hat{\mathbb{E}}_{\mathbb{Q}}\left[\frac{\mathbb{P}\left(\beta^i_{t+1} \mid \beta^j_{0:t}, H^-_{t+1}\right)}{\mathbb{Q}\left(\beta^i_{t+1} \mid \beta^j_{0:t}, H_0\right)}\lambda^j_t\hat{\mathbb{E}}_{z_{t+1}|\beta_{0:t+1},H^-_{t+1}}[\hat{\mathcal{R}}_{t+1}]\right] \tag{3.13}$$

Lemma 3.4.5. *Given an unbiased reward estimator, $\hat{\mathcal{R}}$, the value-function estimator used in HB-MCP is unbiased.*

Proof. Applying similar steps from the proof of lemma A.3.1 on $\hat{\alpha}_{t+1}$, leads to an unbiased value, α_{t+1}. Continuing recursively on the value function yields the desired result. See the appendix for further details. ■

3.5 Negative Information in Ambiguous Data Association

Just like observations affect the hypotheses' weights, not receiving an expected observation also affects the weights, commonly known as negative information. We build on previous work [40] which addresses hybrid Bayesian inference for ambiguous DA and shows how the mathematical formulation naturally extends to include negative information. We limit our discussion of negative information to the context of landmark-based

observations. We conjecture that this formulation can also be adapted to arbitrary observations, but is out of the scope of this work.

Negative information is based on not receiving an observation from a mapped landmark. We denote $|L_t| \in \mathbb{N}$ as the number of mapped landmarks at time instant t. This usually refers to the number of landmarks that already exist in the agent state (but can be defined otherwise). We also define observation as, $z_t = [z_t^1, ...z_t^{|L_t|}]$. Note that there are $|L_t|$ observation elements in the observation, even though usually not all landmarks can be observed at a single time step, as some might be out of the sensing range due to limited field of view, occlusions, and so on. If at time t only $n_{z_t} < |L_t|$ landmarks are observed, we fill the rest of the observation array with $z_t^k = \infty$, i.e., out of sensing range. Then, the observation array becomes $z_t = [z_t^1, ..., z_t^{n_{z_t}}, \infty, ..., \infty]_{1\times|L_t|}$. The reason for such uncommon inflation of the observation array will become clear shortly.

We define $\beta_t = [\beta_{t,1}, ..., \beta_{t,|L_t|}]$ as an array that subscribes each landmark with some observation. For example, $\beta_{t,k} = 1$ associates landmark l^k with observation-element z_t^1 from z_t. Note that by the definition of the observation array, $z_t^{\beta_{t,k}} = \infty$ for all $\beta_{t,k} > n_{z_t}$, which does not correspond to any real observation.

Equipped with the definitions of β_t and z_t, we now discuss the adaptation of the observation and association models. We drop the $\square^{i,j}$ notation to avoid notation overloading, the derivations below are true for each hypothesis separately. In the landmark-based context, it is common to further simplify the expression in (3.4) by assuming conditional independency of an observation given the state variables, to a product of observation models, $\mathbb{P}(z_t \mid X_t, \beta_t) = \prod_{k=1}^{|L_t|} \mathbb{P}(z_t^{\beta_{t,k}} \mid x_t^r, l^k)$, where x_t^r and l^k are the current pose of the agent and landmark k. For simplicity, we assume in this work an ideal detection sensor, in the sense that if a landmark is within range, the sensor will detect it. Under this assumption, likelihood of obtaining an out-of-range observation ($z_t^{\beta_{t,k}} = \infty$), given that the landmark is within the sensing range (denoted $S.R.$), is $\mathbb{P}(z_t^{\beta_{t,k}} = \infty \mid x_t^r, l^k \in S.R.) = 0$. However, obtaining an out-of-range observation given that the landmark is indeed out of the sensing range, is $\mathbb{P}(z_t^{\beta_{t,k}} = \infty \mid x_t^r, l^k \notin S.R.) = 1$.

The association model, $\mathbb{P}(\beta_{t,k} \mid x_t^k, l^k)$, assigns a probability to associate a landmark, l^k, with a specific observation index, $\beta_{t,k}$. We define the likelihood of associating an out-of-sensing-range landmark to an actual observation element (i.e. $\beta_{t,k} \leq n_{z_t}$), as $\mathbb{P}(\beta_{t,k} \leq n_{z_t} \mid x_t^k, l^k \notin S.R.) = 0$. Conversely, associating a landmark that is within the sensing range, equals a nonzero value, for simplicity defined here as a uniform distribution across all feasible associations, $\frac{1}{n_{z_t}}$. We explicitly state all possible combinations of state, association, and observation in table 3.1.

3.6 Experiments

In this section we evaluate our approach, HB-MCP, considering multiple hypotheses due to ambiguous DA. We compare our approach with the state of the art algorithms, DA-

Algorithm	Hypotheses control	Estimator
vanilla-HB-MCTS (3.2.1)	pruning	biased
PFT-DPW [48]	single hypo.	biased
DA-BSP [40]	pruning	biased
HB-MCP **(ours)**	sampling	unbiased

Table 3.2: Algorithms examined in our experiments.

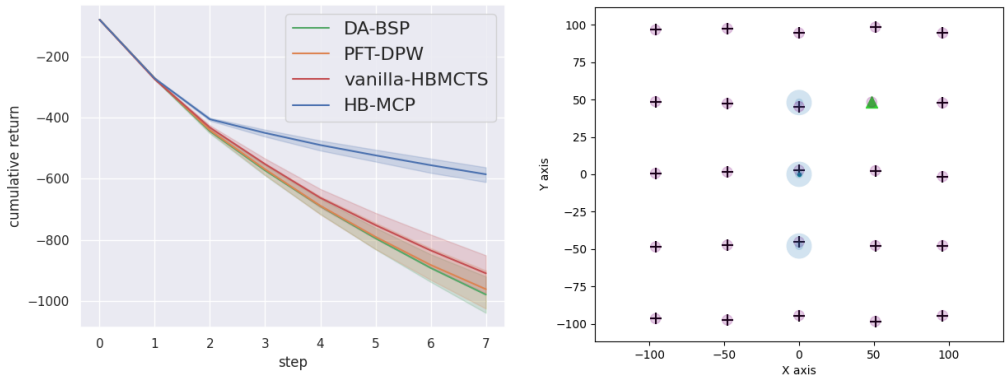
BSP [40] and PFT-DPW [48]. PFT-DPW is utilized here as a single hypothesis solver, as it does not explicitly support multiple hypotheses beliefs. Its hypothesis is chosen based on the hypotheses weights through sampling. While it is possible to modify PFT-DPW to accommodate multiple hypotheses, we leave this for future research. To make DA-BSP comparable to other algorithms, we adapted the algorithm to support anytime planning by utilizing Monte-Carlo trajectory samples instead of a full tree traversal. We also evaluated vanilla-HB-MCTS as the ad-hoc baseline for MCTS implementation with hybrid beliefs, see table 3.2. A summary of the performance of each algorithm is given in table A.3.

In all cases, the experiments were done using GTSAM library [9] with a python wrapper as an inference engine for each of the hypotheses. Most current state-of-the-art online tree search planners rely on particle filters as an inference mechanism. However, particle filters are limited in their ability to support high-dimensional and correlated state spaces efficiently. Instead, through GTSAM we modeled each conditional belief as nonlinear state space model corrupted with multivariate Gaussian noise. We give more information of the hyperparameter choice in the appendix A.3. In the experiments, we assumed a SLAM setting, in which the map is not perfectly known, and the agent is only given a noisy prior on the map and its own pose. Due to ambiguous data associations, each measurement may be obtained from any of the surrounding landmarks within the sensing range of the agent. As a result of the ambiguous data associations, the full posterior belief becomes multi-modal, with discrete variables representing different possible associations.

Aliased matrix. The first environment is a highly aliased map, depicted in figure 3.1(b). The task of the agent is to reduce the uncertainty of its pose and all landmarks of the map, measured by the (negative-) $\mathcal{A}$-optimality criteria. The $\mathcal{A}$-optimality is the trace for the belief covariance matrix, commonly used as uncertainty measure. The state of the agent is its trajectory and prior landmarks. The agent is initially given three

	Aliased matrix	Goal reaching	Kidnapped robot
HB-MCP (ours)	-585.2	-716.8	-323.7
vanilla-HB-MCTS	-909.6	-939.4	-349.5
PFT-DPW	-961.8	-1009.8	-327.8
DA-BSP	-979.5	-931.5	-330.4

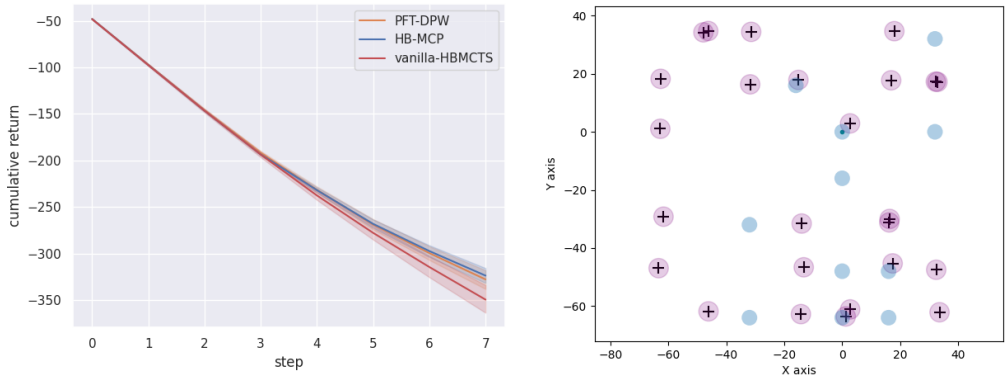
Table 3.3: Comparison of algorithm performances on different scenarios. Results are based on a simulation study with 100 trials per scenario and algorithm.



(a) (b)

Figure 3.1: *Aliased matrix.* The goal of the agent is to minimize the uncertainty of its pose and the location of all landmarks. (a) Mean and standard deviation of the cumulative reward, over 100 trials (higher is better). (b) Illustration of the initial belief of the agent. x^* denotes the ground truth pose of the agent. l^* denotes a unique landmark. The agent receives as a prior three hypotheses at different locations, drawn as blue ellipses.

possible hypotheses for its pose, and 24 aliased landmarks evenly scattered across the map and a unique landmark, given as noisy prior to the agent. The unique landmark breaks the symmetry and may be used by the agent to disambiguate hypotheses. The action space is defined as a straight 4-directional open-loop actions, consisting of 12 intermediate steps, each of 4[m]. Each planning session was limited to 40 seconds.

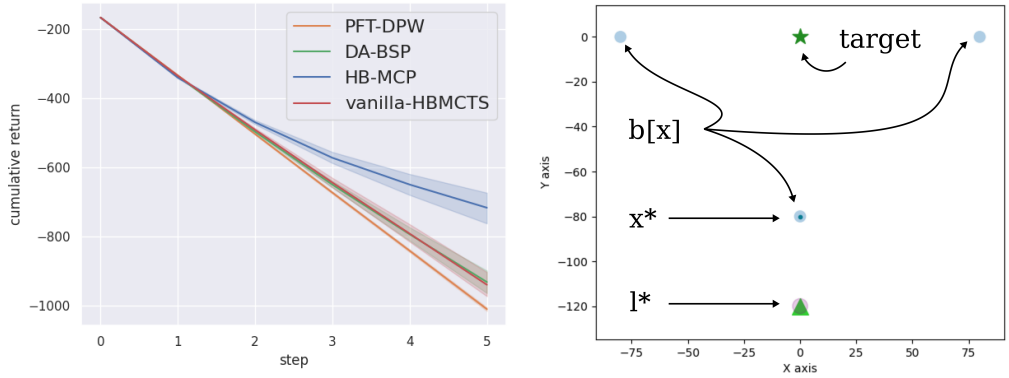


(a) (b)

Figure 3.2: *Kidnapped robot.* The goal of the agent is to minimize the uncertainty of its pose. (a) Mean and standard deviation of the cumulative reward, over 100 trials. (b) Illustration of the initial belief of the agent, blue circles illustrate conditional beliefs, crosses denote landmarks.

Kidnapped robot. The goal of the agent is to minimize the uncertainty about the agent's pose. The environment has 16 randomly scattered landmarks on a $160m \times 160m$ grid, with added Gaussian noise given as prior. The prior pose of the agent is three hypotheses randomly scattered within the grid boundaries. The action space is defined

similarly to aliased matrix environment. The reward function is defined by the $\mathcal{A}$-optimality criteria on the robot's pose. Each planning session was limited to 20 seconds.



(a) (b)

Figure 3.3: *Goal reaching.* The goal of the agent is to reach the target location while minimizing uncertainty. (a) Mean and standard deviation of the cumulative reward, over 100 trials. (b) Illustration of the initial belief of the agent. x^* denotes the ground truth pose of the agent. l^* denotes a unique landmark. The agent receives as a prior three hypotheses at different locations.

Goal reaching. The goal of the agent is to reach a predefined target region. The agent prior belief is given as three hypotheses, located at different directions with respect to the target. To ensure that the right hypothesis gets to the target, the agent must first disambiguate some of the hypotheses (using the unique landmark shown in figure 3.3), and only then attempt to reach the goal. The reward function is defined as the negative sum of the Euclidean distance to goal and the $\mathcal{A}$-optimality criteria. Each planning session was limited to 20 seconds.

HB-MCP received the highest expected cumulative reward in both the ambiguous matrix and goal reaching scenarios. Note how in the ambiguous matrix scenario, HB-MCP achieves significant improvement in cumulative reward from step number 2. The reason for that is the agent's ability to spot the unique landmark, which is two open-loop steps away when $t = 0$, see figure 3.1(a). Due to restricted planning time, vanilla-HB-MCTS and DA-BSP fail to identify and utilize the reduction in uncertainty via disambiguation using the unique landmark. In all cases a single-hypothesis PFT-DPW is unaware of the multi-modality of the problem, and has no incentive to prioritize the unique landmark over any other (ambiguous) landmark. In case of PFT-DPW, this statement is true for all the experiments.

In the kidnapped robot scenario the algorithms performed almost equally well, with slight superiority to HB-MCP. Although PFT-DPW is mathematically inaccurate due to the choice of merely a single hypothesis, it enjoys higher inference and planning efficiency which might translate is in some cases to good performance. Although the kidnapped robot reward punishes for high uncertainty, the random scatter of landmarks

and poses did not lead to any strong preference of a single policy for disambiguation, which can be clearly seen from the cumulative reward of all algorithms in figure 3.2 (a). Clearly, depending on the scenario, even a heuristic, single-hypothesis solver might lead to good performance. For more details, please refer to the appendix.

3.7 Conclusions

In this work, we introduced HB-MCP, a novel algorithm to handle the significant increase in computational effort of planning with hybrid beliefs. We showed that current state-of-the-art algorithms rely on an approximation, namely hypotheses pruning, that leads to a biased and inconsistent reward and value function estimate. We proposed and analyzed a different approach, namely HB-MCP, which utilizes sequential importance resampling to converge to the correct value. Additionally, instead of building symmetric hypotheses trees, HB-MCP focuses computations on the promising branches corresponding to the UCB bonus. We demonstrated how HB-MCP could be used for planning in ambiguous scenarios and derived a simple extension to Bayesian inference to handle negative information naturally. Last, we demonstrated our approach in a simulated environment. In our experiments, HB-MCP outperformed the current state-of-the-art hybrid belief space planning algorithms.

Chapter 4

Data Association Aware POMDP Planning with Hypothesis Pruning Performance Guarantees

In the previous chapter we have considered a hybrid belief that contains both continuous and discrete random variables. We have shown how to use Monte-Carlo approach that both simplifies the computation to improve efficiency, and leads to a mathematically sound approximation that converges to the theoretical solution. In this chapter, we consider a similar setting, where ambiguous data associations lead to a mixture distribution. We show how to simplify calculations while having a bound on the gap between the simplified and non-simplified solutions. Unlike in the previous paper, here we show how to calculate this bound in practice.

We start with some motivation about why should we care about data-associations and specifically data-associations while planning. Clearly, an autonomous agent must reason about partial observability when interacting with the real world. For instance, an autonomous vehicle has to reason about uncertain and incomplete information from its sensors to make decisions such as choosing the correct lane or changing speed. Nevertheless, most planning literature assumes complete knowledge of the source of the observation, i.e., the observed environmental instance, but this may not be true in practice. For example, self-driving cars use camera sensors to observe the scene and relate surrounding objects to an a-priori known map. When a car approaches a controlled intersection, it has to determine which of the visible traffic lights correspond to the traffic light in the map and subsequently apply to the lane it is driving. This is a simple problem if the localization is perfect. However, sensor noise, changing lighting conditions, and occlusions can cause the car to associate observations with an incorrect traffic light. Ignoring the possibility of inconsistent observation associations could lead to an erroneous distribution shift of the state and potentially fatal consequences.

Figure 4.1 provides an example of a robot attempting to reach a destination, rep-

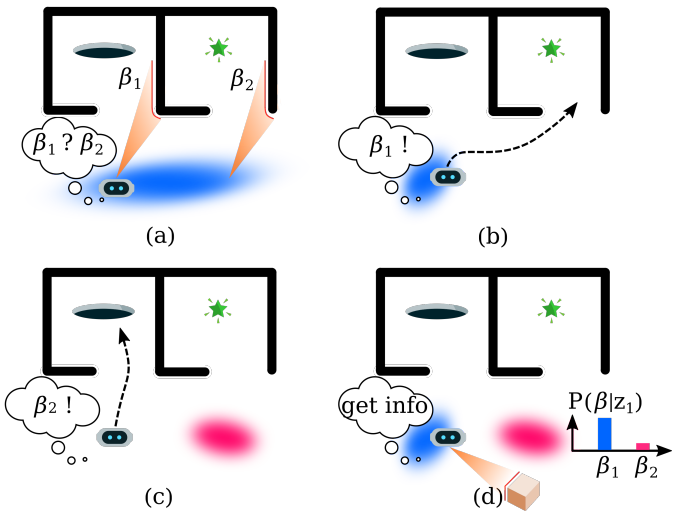


Figure 4.1: Figure (a) depicts an agent aiming to reach a goal (green star) while receiving an observation that could come from two sources, β_1 or β_2. In Figures (b) and (c), incorrect assumptions about the origin of the observation lead to changes in the robot's belief (blue and pink ellipses) and the optimal action, which can vary significantly. Notably, in (c), the calculated best action results in unsafe states. Instead, figure (d) showcases a data association aware belief and action, in which the agent holds two distinct hypotheses. Consequently, the agent chooses an action to gather information rather than traveling directly towards the goal.

resented as a star. In Figure 4.1(a), the robot perceives a potential future observation, but its exact pose is unknown and expressed as a unimodal distribution. Equipped with a sensor having a limited field of view, the robot detects a portion of a wall, which could be part of a corridor leading to the goal (high reward) or a pit (low reward). In Figures 4.1(b) and (c), the robot assumes a deterministic source for the observation, leading to potential selection of an incorrect and possibly unsafe action. Figure 4.1(d) demonstrates a multi-modal posterior belief with different data association possibilities. Consequently, the agent decides to gather more information rather than directly moving toward the goal. This example highlights the importance of accounting for data association ambiguity to avoid poor performance and unsafe policies where the agent might mistakenly head towards the pit instead of the star.

In general POMDPs, a plan that accounts for uncertainty maintains a distribution over the possible states of the world. Accounting for ambiguous data associations adds another layer of complexity by having to consider multiple hypotheses, leading to a mixture distribution, where each component of the mixture corresponds to a single hypothesis. Additionally, as the planning horizon grows, the number of hypotheses grows exponentially [40], adding a significant computational burden.

In response to the challenges posed by ambiguous data associations in POMDPs, we propose a simplification approach, which maintains a small subset of the hypotheses instead of maintaining an exponential number thereof. Importantly, we derive bounds on the utility function between the POMDP with the simplified and the non-simplified beliefs. We use these bounds to establish a trade-off between computational efficiency and performance for state-dependent rewards. Further, using this relationship, we propose a novel pruning approach that balances computational efficiency with performance

loss by adaptively selecting which hypotheses to prune online.

Unlike current state-of-the-art POMDP planners that rely on particle propagation, e.g. POMCP or DESPOT, our proposed approach overcomes the challenge of particle depletion by introducing a novel estimator for the objective function. This estimator is agnostic to the inference mechanism being used, it supports both nonparametric and parametric inference mechanisms to enable long planning horizons. Through experiments in simulated environments, we demonstrate the effectiveness of our proposed approach in handling multi-modal belief hypotheses with ambiguous data associations.

In this chapter we make the following main contributions: (a) we derive a theoretical relation between the POMDP with a complete set of hypotheses and the pruned set of hypotheses, enabling us to establish a trade-off between computational efficiency and performance; (b) we develop an estimator that enables parametric and nonparametric belief mixture representation to address particle depletion; (c) we establish a similar relation between an estimated value function based on the complete set of hypotheses and the value function of the pruned set of hypotheses; (d) our bounds can be utilized to provide guarantees in terms of worst-case loss in planning performance given some pruning method; (e) moreover, we derive a scheme that utilizes our bounds to adaptively decide which hypotheses to prune to meet a user-defined allowable loss in planning performance. Finally, we demonstrate the effectiveness of our planning algorithm in a simulated environment with unresolved data associations leading to multi-modal belief.

4.1 Preliminaries

The reward is defined as an expectation over a state-dependent function, $r(b_t, a_t) = \mathbb{E}_{x\sim b_t}[r_x(x, a_t)]$. $\mathcal{R}_{max}$ denotes the maximal value of the reward function, $\mathcal{R}_{max} = \arg\max_{x\in\mathcal{X}, a\in\mathcal{A}}\{r_x(x, a)\}$. The value function for a policy π over a finite horizon $\mathcal{T}$ is defined as the expected cumulative reward received by executing π,

$$V^{\pi}(b_t) = r(b_t, \pi_t) + \mathop{\mathbb{E}}_{z_{t+1:\mathcal{T}}}\left[\sum_{\tau=t+1}^{\mathcal{T}} r(b_\tau, \pi_\tau)\right]. \tag{4.1}$$

The action-value function is defined by executing action a_t and then following policy π for a finite horizon $\mathcal{T}$. The goal of the agent is to find the optimal policy π^* that maximizes the value function. In the rest of the chapter we write $\pi_t \equiv \pi(b_t)$ for conciseness.

4.1.1 Ambiguous Data Associations as Mixture Belief

To represent ambiguous data associations within the POMDP framework we define the belief as a mixture distribution, that encompasses both continuous and discrete random variables. The discrete variables, β_t, represent different associations to seen

observations at time t. We formally define the mixture belief at each time t as,

$$b\left(x_t\right) = \sum_{\beta_{0:t}} \mathbb{P}(\beta_{0:t} \mid H_t)\mathbb{P}(x_t|\beta_{0:t}, H_t), \tag{4.2}$$

where $\mathbb{P}(\beta_{0:t} \mid H_t)$ is the marginal belief over discrete variables which can be considered as the mixture weight. An hypothesis, $\beta_{0:t}$, denote the entire sequence of associations up to time step t. $\mathbb{P}(x_t|\beta_{0:t}, H_t)$ is the conditional belief over continuous variables, given that the history and associations are known. The marginal belief over the hypothesis, $\beta_{0:t}$, can be updated by applying Bayes rule followed by chain rule,

$$\begin{aligned} \mathbb{P}(\beta_{0:t} \mid H_t) &= \eta_t \mathbb{P}(z_t \mid \beta_{0:t}, H_t^-)\mathbb{P}(\beta_{0:t} \mid H_t^-) \\ &= \eta_t \mathbb{P}(z_t \mid \beta_{0:t}, H_t^-)\mathbb{P}(\beta_t \mid \beta_{0:t-1}, H_t^-)\mathbb{P}(\beta_{0:t-1} \mid H_t^-). \end{aligned} \tag{4.3}$$

The conditional belief is updated for each realization of discrete random variables as

$$\mathbb{P}(x_t|\beta_{0:t}, H_t) = \psi\big(\mathbb{P}(x_{t-1}|\beta_{0:t-1}, H_{t-1}), a_{t-1}, z_t\big), \tag{4.4}$$

where $\psi(.)$ represents the Bayesian inference method. Last, the reward function can now be written in terms of hypothesis dependency, $r(b_t, a_t) = \mathbb{E}_{x\sim b_t}[r_x(x, a_t)] = \mathbb{E}_{\beta_{0:t}}\left[\mathbb{E}_x[r_x(x, a_t) \mid \beta_{0:t}]\right]$. For conciseness, we will denote

$$r(b_t^\beta, \pi_t) \triangleq \mathbb{E}_x[r_x(x, a_t) \mid \beta_{0:t}]. \tag{4.5}$$

4.1.2 IS and SN estimators

Importance sampling (IS) is a Monte Carlo simulation technique for estimating the expected value of a target function with respect to a probability distribution. The IS estimator involves drawing samples from a proposed distribution and weighting them by the ratio of the target distribution, $\mathbb{P}(\cdot)$ to the proposal distribution, $Q(\cdot)$,

$$\hat{\mathbb{E}}^{IS}\left[r_x(x)\right] \triangleq \frac{1}{N}\sum_{i=1}^{N} \omega(x^i) r_x(x^i) = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathbb{P}(x^i)}{Q(x^i)} r_x(x^i). \tag{4.6}$$

The estimator is unbiased and consistent [12], when the proposal distribution is non-zero wherever the target distribution is non-zero. Self-normalized importance sampling sometimes serves as a lower-variance estimator by normalizing the importance weights. The SN-estimator is described as,

$$\hat{\mathbb{E}}^{SN}\left[r_x(x)\right] \triangleq \sum_{i=1}^{N} \frac{\omega(x^i)}{\sum_{j=1}^{N} \omega(x^j)} r_x(x^i), \tag{4.7}$$

which converts the weights to a probability distribution. The SN-estimator is biased, but consistent estimator.

4.2 Planning with Ambiguous Data Associations

In this section, we provide an overview of our algorithm, DA-MCTS, and the baseline algorithm, vanilla Hybrid Belief-MCTS (HB-MCTS) [3]. To facilitate understanding, we present the pseudo-code for both algorithms jointly in Algorithm 4.1. We adopt a unified view, with comments indicating the lines unique to each algorithm.

DA-MCTS is built upon the vanilla HB-MCTS algorithm, which itself is an adaptation of PFT-DPW [48] and MCTS [27]. While we have chosen to use these algorithms as the foundation for our work, we acknowledge that other approaches may also be applicable, and we leave exploration of these avenues to future research.

Vanilla HB-MCTS, a variant of belief-Markov Decision Process (BMDP), reframes the POMDP into a belief-state model. In this, states are replaced by belief-states reflecting an agent's environmental uncertainty. The transition and observation functions update prior to posterior beliefs based on action and observation, mirroring the stochastic state changes in a standard MDP. By transforming POMDP to a BMDP, many MDP planning algorithms, including MCTS, can be used as planning solvers. Notably, single particle propagation algorithms, such as POMCPOW, are also possible, but may suffer from particle depletion as mentioned in section 4.

Algorithm 4.1 presents a pseudo-code for the vanilla HB-MCTS algorithm. In the SIMULATE procedure, an action is selected based on the Upper Confidence Bound (UCB) heuristic in line 4. Depending on whether the budget on the number of observations has been met, the algorithm either expands a new posterior node, which includes its belief and reward function, and then performs a rollout, or uniformly samples an existing posterior node and continues recursively to the next node. Finally, the action value of the current node and its relevant counters are updated. The vanilla HB-MCTS algorithm is flexible in that the number of maintained posterior hypotheses can be controlled and remain fixed based on a pre-defined hyperparameter. For instance, a vanilla HB-MCTS with low compute resources can have a pruning budget, where only K hypotheses are maintained in each node of the planning tree. The pruned hypotheses are usually chosen heuristically, e.g. based on their probability value.

However, Vanilla HB-MCTS is limited in its ability to provide guarantees when pruning is performed. While the performance guarantees we present in the next section are applicable to any pruning heuristic, such as the one used in vanilla HB-MCTS, we introduce a slightly different approach. Instead of pre-defining a fixed number of hypotheses to maintain, we propose an adaptive approach that determines which hypotheses to prune online based on a pre-defined maximum allowable loss, $\epsilon_{\bar{D}}$. We then modify the HB-MCTS algorithm to adaptively determine which hypotheses to prune, while maintaining performance guarantees with respect to the complete set of hypotheses. This modification is reflected in line 7.

In addition, DA-MCTS can provide even tighter guarantees in hindsight without incurring additional computational complexity, denoted by $\hat{\epsilon}^{hs}_{\bar{D}}$, shown in line 18. The in-

creased accuracy of these guarantees is due to the granularity of the hypotheses weights. For instance, when there is only a single hypothesis, no hypotheses are pruned, resulting in zero additional loss to the value function. The specific bounds and estimators used are discussed in the following section.

Algorithm 4.1 HB-MCTS and DA-MCTS

Procedure:SIMULATE($b, h, d, \epsilon_{\bar{D}}$)
/*Init: $N(b), N(ba), Q(ba), \hat{\epsilon}^{hs}_{\bar{D}}(b), \hat{\delta}^{\beta}_{\bar{D}}(b)$ to 0*/
if d = 0 **then**
  Return 0
end if
$a \leftarrow \arg\max_{\bar{a}} Q(b\bar{a}) + c\sqrt{\frac{log(N(b))}{N(b\bar{a})}}$
if $|C(ba)| \leq k_o N(ba)^{\alpha_o}$ **then**
  $b' \leftarrow$ PRUNEDPOSTERIOR(b, a) /*Vanilla HB-MCTS*/
  $b', \delta^{\beta}_{\bar{D}} \leftarrow$ PRUNINGWITHGUARANTEES$(b, a, \epsilon_{\bar{D}})$ /*DA-MCTS. Eq. (4.13)*/
  $r \leftarrow$ REWARD(b, a)
  $C(ba) \cup \{(b', r)\}$
  $R \leftarrow r+$ROLLOUT$(b', d-1)$
else
  $b', r \leftarrow$ Sample uniformly from $C(ba)$
  $R,\ \hat{\epsilon}^{hs}_{\bar{D}} \leftarrow r+$SIMULATE$(b', d-1, \epsilon_{\bar{D}})$
end if
$N(b) \leftarrow N(b) + 1$
$N(ba) \leftarrow N(ba) + 1$
$Q(ba) \leftarrow Q(ba) + \frac{R - Q(ba)}{N(ba)}$
$\hat{\epsilon}^{hs}_{\bar{D}} \leftarrow$ GETGUARANTEES$(\hat{\epsilon}^{hs}_{\bar{D}}, \hat{\delta}^{\beta}_{\bar{D}})$ /*DA-MCTS. Eq. (4.12)*/
return $R,\ \hat{\epsilon}^{hs}_{\bar{D}}$

4.3 Mathematical Analysis

In this section, we mathematically analyze the impact of pruning on the performance of the agent. We establish a novel relationship between the complete and pruned value functions for state-dependent reward functions and provide bounds on the loss of approximation. Due to restricted space we defer most proofs and derivations to the appendix, A.4.

We define $D_t = \{\beta_t^1, \beta_t^2, ..., \beta_t^{|D_t|}\}$ the set of associations at time step t, and $\overline{D}_t \subseteq D_t$ as the subset of hypotheses survived after the pruning procedure. We define the pruned belief as,

$$\bar{b}_t \triangleq \bar{\mathbb{P}}(x_t \mid H_t) = \sum_{\beta_t \in \overline{D}_t} \mathbb{P}(x_t \mid \beta_t, H_t)\bar{\mathbb{P}}(\beta_t \mid H_t), \tag{4.8}$$

where the $\bar{\square}$ notation indicates a pruned distribution after normalization. This can be explicitly written as,

$$\bar{b}_t = \int_{x_{t-1}} \bar{b}_{t-1} \frac{\sum_{\beta_t \in \overline{D}_t} \mathbb{P}(z_t \mid x_t, \beta_t)\mathbb{P}(\beta_t \mid x_t)\mathbb{P}(x_t \mid x_{t-1}, \pi_{t-1})}{\overline{\mathbb{P}}\left(z_t \mid H_t^-\right)}, \tag{4.9}$$

where, $\overline{\mathbb{P}}\left(z_t \mid H_t^-\right) = \int_{x_{t-1:t}} \sum_{\beta_t \in \overline{D}_t} \mathbb{P}(z_t \mid x_t, \beta_t)\mathbb{P}(\beta_t \mid x_t)\mathbb{P}(x_t \mid x_{t-1}, \pi(z_{t-1}))\bar{b}_{t-1}$.

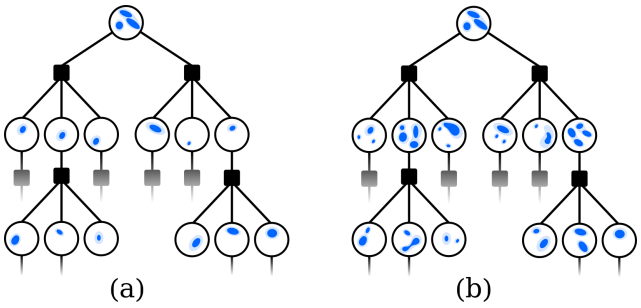


Figure 4.2: Planning trees with nodes representing beliefs, and inner blue shapes illustrate distributions of the conditional posteriors. (a) A belief tree with standard Monte-Carlo estimator leads to an overconfident, fully observed data association after a single step. (b) A planning tree with Self-Normalized Importance Sampling estimators to account for different hypotheses at posterior nodes.

Note that the summation is over the pruned set of hypotheses.

Theorem 4.1. *Let time-step* 0 *denote the root of the planning tree. Then, the expected reward for the pruned POMDP,* $\overline{M}$*, is bounded with respect to the full POMDP,* M*, through the factor of the pruned weight values, and the maximum immediate reward,*

$$\left|\mathbb{E}[r(b_t,a_t)]-\overline{\mathbb{E}}[r(\bar{b}_t,a_t)]\right| \leq \mathcal{R}_{max}\left[\delta_0^\beta + \sum_{\tau=1}^{t-1}\overline{\mathbb{E}}_{z_{1:\tau}}\left[\delta_\tau^\beta\right]\right], \tag{4.10}$$

where $\delta_\tau^\beta \triangleq \sum_{\beta_\tau \in D_\tau \setminus \overline{D}_\tau} \overline{\mathbb{P}}(\beta_\tau \mid H_\tau)$*, i.e. the sum of pruned hypotheses weights at time-step* τ*.*

Crucially, in order to calculate the value of δ_τ^β, the values of the hypotheses weights which are descendent of past pruned hypotheses are not required, as they cannot be obtained without explicitly calculating all hypotheses. More formally, $\overline{\mathbb{P}}(\beta_t \mid H_t) = \frac{\mathbb{P}(z_t|\beta_t,H_t^-)\sum_{\beta_{0:t-1}\in\overline{D}}\mathbb{P}(\beta_t|\beta_{0:t-1},H_{t-1})\mathbb{P}(\beta_{0:t-1}|H_t^-)}{\overline{\mathbb{P}}(z_t|H_t^-)}$ has summation only over the survived hypotheses.

The generalization of theorem A.1 to the entire value function, is straightforward due to linearity of the expectation,

Corollary 4.2. *Without loss of generality, assume that the time step at the root node of the planning tree is* $t = 0$*. Then, for any policy* π*, the following holds,*

$$\left|V^\pi(b_0)-\bar{V}^\pi(\bar{b}_0)\right| \leq \mathcal{R}_{max}\left[\mathcal{T}\delta_0^\beta + \sum_{k=1}^{\mathcal{T}}\sum_{\tau=1}^{k}\overline{\mathbb{E}}_{z_{1:\tau}}\left[\delta_\tau^\beta\right]\right]. \tag{4.11}$$

For conciseness, we denote this bound as $\epsilon_{\overline{D}}^{hs}$. As we will derive in the following sections, an equivalent bound can be derived for estimated value functions, that is,

$$|\hat{V}^\pi(\hat{b}_0) - \hat{\bar{V}}^\pi(\hat{\bar{b}}_0)| \leq \mathcal{R}_{max}\left[\mathcal{T}\hat{\delta}_0^\beta + \sum_{k=1}^{\mathcal{T}}\sum_{\tau=1}^{k}\hat{\overline{\mathbb{E}}}_{z_{1:\tau}}\left[\hat{\delta}_\tau^\beta\right]\right], \tag{4.12}$$

where $\hat{\square}$ denotes an estimator. Similarly, we denote $\hat{\epsilon}_{\overline{D}}^{hs}$ as the (deterministic) bound for the estimated value functions.

4.3.1 Adaptive Pruning with Performance Guarantees

The theoretical value bound in Equation (A.115) and the estimator value bound in Equation (4.12) can be used to provide guarantees for various pruning heuristics, including those presented in prior work such as [40, 3] by providing guarantees after the planning session has ended.

In this section, we go a step further, and propose a novel mechanism for selecting the surviving hypotheses. Unlike previous approaches that use a fixed budget on the number of allowed hypotheses [40], our algorithm requires the user to specify the maximum allowable loss, $\epsilon_{\bar{D}}$, on the value function. Using this allowable loss, our algorithm dynamically selects the cardinality and instances of hypotheses to prune online, while maintaining the performance guarantees provided in advance.

To achieve this, we set the value of $\epsilon_{\bar{D}}$ and by construction determine δ_τ^β to be a constant, denoted as Δ, for all H_τ and all time steps τ. We use Δ to determine which hypotheses to prune in order to meet the budget. The resulting bound can then be expressed as follows,

$$\begin{aligned}\left|V^\pi(b_0) - \bar{V}^\pi(\bar{b}_0)\right| &\leq \mathcal{R}_{max}\Delta\left[\mathcal{T} + \sum_{k=1}^{\mathcal{T}}\sum_{\tau=1}^{k} 1\right] \\ &= \mathcal{R}_{max}\Delta\left[\frac{\mathcal{T}^2 + 3\mathcal{T}}{2}\right] \triangleq \epsilon_{\bar{D}}.\end{aligned} \tag{4.13}$$

The hyperparameter $\epsilon_{\bar{D}}$ controls the maximum allowable loss and is set a priori, as a result Δ can easily be derived. During planning, we sum over δ_τ^β, until its value is as close as possible to Δ without crossing its value. The difference between these two values allows us to obtain a tighter guarantee in hindsight, $\epsilon_{\bar{D}}^{hs}$, which satisfies the inequality $\epsilon_{\bar{D}}^{hs} \leq \epsilon_{\bar{D}}$. A similar claim can be made for the sampling-based bound. The formal derivation of these estimators is presented in the next section.

4.3.2 Estimated expected reward

In this section, we first develop an estimator for the value function, assuming the availability of a complete set of hypotheses at each posterior belief. Then, we derive a similar, pruning-based estimator. In the next section, we will show a deterministic relation between the estimators. However, before delving into the details, we first give a motivation for deriving guarantees with respect to the estimators.

As stated in Corollary A.2, the value function based on the complete set of hypotheses should not deviate significantly from the value function based on the pruned hypotheses set, as long as the pruned hypotheses have low weight values. However, in practice, current state-of-the-art algorithms cannot compute the full nor the pruned value functions due to intractable integrals involved with expectations. Online POMDP algorithms provide performance guarantees based on estimated value functions, where a sampled set of observations and states approximate expectations and the belief dis-

tribution, e.g., [45, 32].

For clarity, we derive the estimator by considering separately each expected reward along the planning horizon. Using linearity of the expectation, the value function may be written as,

$$V^{\pi}(b_0) = r(b_0, \pi_0) + \sum_{t=1}^{\mathcal{T}} \mathbb{E}_{z_{1:t}}[r(b_t, \pi_t)]. \tag{4.14}$$

We handle each term in the summation individually, and make the following proposition as a first step towards deriving an estimated expected reward,

Proposition 4.3.1. *Let $z_{1:t}$ denote an observation sequence, $r(b_t, \pi_t)$ be the reward value for a given belief, b_t and policy π_t. The expected reward value can be written as,*

$$\mathbb{E}_{z_{1:t}}[r(b_t, \pi_t)] = \tag{4.15}$$
$$\int_{z_{1:t}} \mathbb{E}_{\beta_0} \prod_{\tau=1}^{t} \mathbb{E}_{\beta_\tau | \beta_{0:\tau-1}} \left[\mathbb{P}\left(z_\tau \mid \beta_{0:\tau}, H_\tau^-\right) r\left(b_t^\beta, \pi_t\right) \right],$$

where $r\left(b_t^\beta, \pi_t\right)$ denotes the reward value of a single hypothesis realization, $\beta_{0:t}$, as shown in equation (4.5).

From the proposition we derive a standard Monte-Carlo sampling approach, where we iteratively sample sequences of hypotheses $\beta_{0:t}$ and observation samples, $z_{1:t}$,

$$\hat{\mathbb{E}}_{z_{1:t}}^{MC}[r(\hat{b}_t, \pi_t)] = \frac{1}{N} \sum_i \hat{r}\left(b_t^{\beta^i}, \pi_t\right), \tag{4.16}$$

where $\square^{MC}$ denotes Monte-Carlo estimation and $b_t^{\beta^i} \triangleq \mathbb{P}\left(x_t \mid \beta_{0:t}^i, z_{1:t}^i, \pi_{0:t-1}\right)$. However, since the observation space is continuous, different realizations of $\beta_{0:t}$, denoted $\beta_{0:t}^i$, will never sample the same observation sequence $z_{1:t}^i$ twice. In the planning tree, it means that after an observation sample, there is only a single hypothesis in any posterior node, resulting in a fully observed data association. However, if the agent obtains an observation in the real world, the data association ambiguity is generally not fully resolved. A result, the Monte Carlo sampling approach is an over-optimistic, erroneous planner which only considers ambiguity at the root node of the planning tree. See figure 4.2 for an illustration.

Inspired by [48] for standard POMDPs, and [3] for hybrid POMDPs, we derive an Importance Sampling (IS) estimator, which may sample observations from different distributions, and weigh each hypothesis with an importance weight, $\omega\left(z_\tau\right)$. The importance weight reflects the probability of observing z_t given hypothesis $\beta_{0:\tau}$ and history H_τ^-, normalized to the actual sampling distribution being used, $Q(\cdot)$. We may

write equation (4.15) to reflect the change,

$$\mathbb{E}_{z_{1:t}}[r(b_t,\pi_t)] = \tag{4.17}$$
$$\int_{z_{1:t}} \mathbb{E}_{\beta_0} \prod_{\tau=1}^{t} Q\left(z_\tau \mid H_\tau^-\right) \mathbb{E}_{\beta_\tau|\beta_{0:\tau-1}} \left[\omega\left(z_\tau\right) r\left(b_t^\beta, \pi_t\right)\right]$$

where $\omega\left(z_\tau\right) = \frac{\mathbb{P}\left(z_\tau|\beta_{0:\tau}, H_\tau^-\right)}{Q\left(z_\tau|H_\tau^-\right)}$ and $Q(.)$ is the proposal distribution from which the sampling-based estimator will sample observations. Clearly, the two terms are equivalent. From (4.17) we can directly derive the IS-estimator,

$$\hat{\mathbb{E}}^{IS}_{z_{1:t}}[r(\hat{b}_t)] = \hat{\mathbb{E}}_{z_{1:t}}\mathbb{E}_{\beta_{1:t}}[\hat{r}\left(b_t^\beta, \pi_t\right)] \triangleq \tag{4.18}$$
$$\sum_{z^c_{1:t}} \sum_{\beta_{0:t} \in D_{0:t}} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega\left(z_\tau^c\right)}{N} \hat{r}\left(b_t^\beta, \pi_t\right),$$

where, $\hat{r}(b_t^\beta, \pi_t)$ is the sample-based mean for the state-reward over the conditional belief, as defined in equation (4.5). In contrast to the standard Monte-Carlo estimator (4.16), using an importance sampling estimator enables us to reason about all hypotheses for every observation sequence, shown by the summation over $\beta_{0:t}$ for each sampled $z^c_{1:t}$.

Although the IS estimator is theoretically justified as a consistent and unbiased estimator, we make another step in deriving the estimator and use a Self-Normalized Importance Sampling (SN) estimator,

$$\hat{\mathbb{E}}^{SN}_{z_{1:t}}[r(\hat{b}_t)] = \hat{\mathbb{E}}_{z_{1:t}}\mathbb{E}_{\beta_{1:t}}[\hat{r}\left(b_t^\beta, \pi_t\right)] \triangleq \tag{4.19}$$
$$\sum_{z^c_{1:t}} \sum_{\beta_{0:t} \in D_{0:t}} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}(\beta_\tau|\beta_{0:\tau-1}, H_\tau^-) \frac{\omega\left(z_\tau^c\right)}{\sum_{z_\tau^k} \omega\left(z_\tau^k\right)} \hat{r}\left(b_t^\beta, \pi_t\right)$$

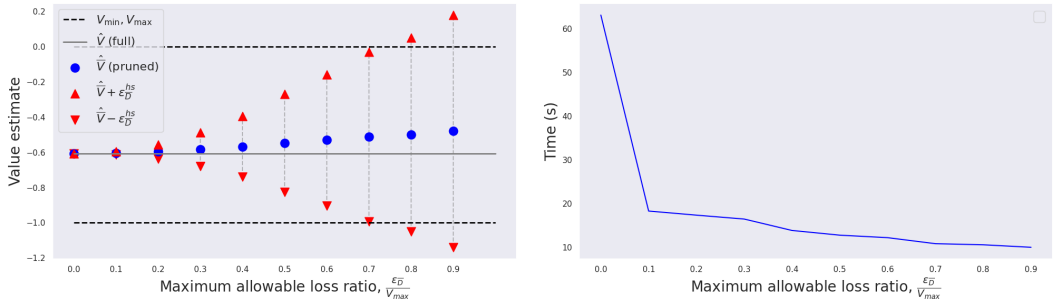
The SN-estimator is no longer unbiased, but is known to be consistent [12]. The main reason for that step is to achieve a bounded deterministic difference between the full and pruned estimators, as we will describe in the following section.

Last, we derive a similar estimator for the *pruned* posterior belief,

$$\hat{\bar{\mathbb{E}}}_{z_{1:t}}\left[r\left(\hat{\bar{b}}_t, \pi_t\right)\right] = \hat{\mathbb{E}}_{z_{1:t}}\bar{\mathbb{E}}_{\beta_{1:t}}[\hat{r}\left(b_t^\beta, \pi_t\right)] \triangleq \tag{4.20}$$
$$\sum_{z^c_{1:t}} \sum_{\beta_{0:t} \in \overline{D}_{0:t}} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-) \frac{\omega\left(z_\tau^c\right)}{\sum_{z_\tau^k} \omega\left(z_\tau^k\right)} \hat{r}\left(b_t^\beta, \pi_t\right).$$

4.3.3 Estimators analysis

In this section, we derive a bounded relationship between the full and pruned estimators. Finally, we discuss how these estimators relate to the theoretical value function.



(a) (b)

Figure 4.3: (a) Bounds of our approach with respect to level of simplification. $\hat{V}, \hat{\bar{V}}$ are the value functions of the full and pruned estimators respectively. $\hat{\bar{V}} + \hat{\epsilon}_{\bar{D}}^{hs}$ represent the bounds of the pruned estimator. $V_{\min}, \backslash V_{\max}$ represent the minimum and maximum theoretical values of the value function. All values are normalized with respect to $\max\{|V_{min}|, |V_{max}|\}$. Here $|V_{max}| \equiv 0$ since the reward is defined as the negative Euclidean distance to goal. (b) Time for task completion with respect to level of simplification. Each level corresponds to the bounds presented in figure (a).

Theorem 4.3. *Let π be a policy, then the expected reward for the estimated pruned POMDP, $\hat{\overline{M}}$, is bounded with respect to the estimated full POMDP, $\hat{M}$, as follows,*

$$\left|\hat{\mathbb{E}}^{\pi}_{z_{1:t}}[r(\hat{b}_t)] - \hat{\bar{\mathbb{E}}}^{\pi}_{z_{1:t}}\left[r\left(\hat{\bar{b}}_t\right)\right]\right| \leq \mathcal{R}_{max}\left[\hat{\delta}_0^{\beta} + \sum_{\tau=1}^{t}\hat{\delta}_{\tau}^{\beta}\right]. \tag{4.21}$$

where, $\hat{\delta}_{\tau}^{\beta} = \hat{\mathbb{E}}_{z_{1:t}^{c}}\overline{\mathbb{E}}_{\beta_{0:t-1}}\sum_{\beta_t \in D_t \backslash \overline{D}_t}\mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^{-}\right)$ for all $\tau \in [1,t]$ represents the expected sum of conditional hypotheses' weights which are myopically pruned and $\hat{\delta}_0^{\beta} = \sum_{\beta_0 \in D_0 \backslash \overline{D}_0}\mathbb{P}\left(\beta_0 \mid H_t^{-}\right)$.

In accordance with the theoretical case, as described in Equation (4.17), to evaluate $\hat{\delta}_{\tau}^{\beta}$, only the surviving hypotheses from past time steps are needed. The theorem can be generalized to the full value function by re-introducing the summation. Under the assumptions of theorem A.3 the following holds,

Corollary 4.4. *The difference between the estimated value function of the full POMDP, $\hat{M}$, and the estimated value function of the pruned POMDP, $\hat{\overline{M}}$, is bounded by,*

$$|\hat{V}^{\pi}(\hat{b}_0) - \hat{\bar{V}}^{\pi}(\hat{\bar{b}}_0)| \leq \mathcal{R}_{max}\left[\mathcal{T}\hat{\delta}_0^{\beta} + \sum_{k=1}^{\mathcal{T}}\sum_{\tau=1}^{k}\hat{\delta}_{\tau}^{\beta}\right]. \tag{4.22}$$

The corollary relates the complete but computationally expensive value function estimator to the efficient, pruning-based estimator. Both estimators utilize the same sampled observations since they share the same proposal distribution.

Finding a finite sample algorithm with practical guarantees between the estimated value function and the theoretical remains an open challenge in the POMDP literature and is aside from our current contribution. Nevertheless, to fully justify our approach, we formally state that given such an algorithm, denoted $\mathcal{A}$, that utilizes the importance sampling estimator defined in equation (4.19), our simplified estimator provides a relationship to the theoretical value function while being more efficient,

Corollary 4.5. *Let π be a policy and let $\mathcal{A}$ be a sampling-based estimator for the value function such that $|V^{\pi}(b_0) - \hat{V}^{\pi}(\hat{b}_0)| \leq \epsilon_{\mathcal{A}}$ with probability at least $1 - \delta_{\mathcal{A}}$. Then, the loss in the value function for the pruned hypotheses is bounded,*

$$|V^{\pi}(b_0) - \hat{\bar{V}}^{\pi}(\hat{\bar{b}}_0)| \leq \tag{4.23}$$

$$|V^{\pi}(b_0) - \hat{V}^{\pi}(\hat{b}_0)| + |\hat{V}^{\pi}(\hat{b}_0) - \hat{\bar{V}}^{\pi}(\hat{\bar{b}}_0)| \leq \epsilon_{\mathcal{A}} + \hat{\epsilon}^{hs}_{\bar{D}}, \tag{4.24}$$

and holds with probability $1 - \delta_{\mathcal{A}}$. We use $\hat{\epsilon}^{hs}_{\bar{D}}$ as a shorthand for the bounds provided in corollary A.4.

The results established so far hold for any policy, assuming that both the theoretical and estimated value functions are based on the same policy. However, planning based on the pruned belief may result in a different policy from the optimal one for the underlying POMDP. Nevertheless, we demonstrate that the optimal policy for the pruned and potentially sampled-based POMDP, denoted $\bar{\pi}$, incurs bounded loss in performance compared to the optimal policy for the full theoretical POMDP, denoted $\pi^{\star}$.

Corollary 4.6. *Let $\bar{\pi}$ be the optimal policy for the pruned, possibly sampled-based POMDP and $\pi^{\star}$ be the optimal policy for the full theoretical POMDP. Then,*

$$\left| V^{\pi^{\star}}(b_t) - \hat{\bar{V}}^{\bar{\pi}}(\hat{\bar{b}}_t) \right| \leq 2 \left(\epsilon_{\mathcal{A}} + \hat{\epsilon}^{hs}_{\bar{D}} \right). \tag{4.25}$$

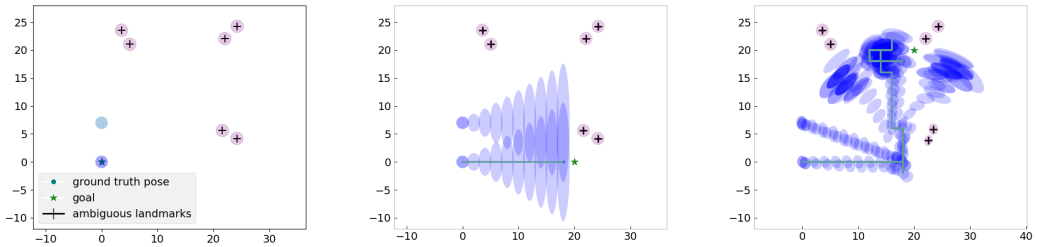
This is an unsurprising result, since the best policy for the pruned approximation, $\bar{\pi}$, should perform no worse than the optimal policy, $\pi^{\star}$, for the simplified POMDP or otherwise it would have been selected.

4.4 Experiments

In this section we experiment with different pruning approaches to validate our findings. We use MCTS as a baseline algorithm and compare multiple hypothesis pruning approaches to our adaptive scheme. The experimental evaluation of our approach consists of two main parts. In the first part, we validate the proposed bounds and investigate their sensitivity to the level of simplification chosen. In the second part, we conduct a simulation study to demonstrate the practical performance gains of our adaptive pruning approach.

Importantly, we emphasize that the theoretical guarantees presented in section 4.3 are suitable for other hypotheses-based algorithms as well, such as [40, 3] or PFT-DPW [48] if the latter is adapted to multiple hypotheses.

To conduct the simulations, we utilized the GTSAM library [9] as our inference engine. Our belief model is based on a Gaussian Mixture Model, in which each posterior belief in the planning tree corresponds to multiple instances of GTSAM factor graphs.



(a) (b) (c)

Figure 4.4: The figures demonstrate the estimated state of the entire trajectory, also known as the smoothing state, of the agent at time t given the observed history. (a) The prior of the agent given as two Gaussian hypotheses. Each Gaussian is represented as an ellipse illustrating its covariance, centered around its mean. The landmarks are part of the agent state a priori but have an uncertain location, with ellipses illustrating their covariances. (b) The belief of the agent adjacent to the first waypoint before obtaining any observation. (c) The belief of the agent after pruning. Non-negligible hypotheses differ substantially.

Each instance represents a conditional posterior over the continuous part of the belief, $\mathbb{P}(x_t \mid \beta_{0:t}, H_t)$, while the discrete part of the belief, $\mathbb{P}(\beta_{0:t} \mid H_t)$, is maintained as a list of probability values, each corresponds to an hypothesis. Apart from the pruning method, which is the focus of this section, all hyper-parameters are shared across all solvers and remain fixed. The planning is performed in a receding horizon manner, where after each planning session, only the first action is executed, and all calculations are done from scratch in the subsequent step.

In the first experiment the belief of the agent included the pose of the agent and two ambiguous landmarks. The objective of the agent was to reach a target destination, encoded into the reward function as the expected Euclidean norm between the agent pose samples and the target. The field of view of the agent was chosen to be unbounded and with unlimited sensing range, that is, at every time step, the agent obtains an observation from two sources, but cannot identify its source. In this simple toy example, the number of hypotheses quickly grows and becomes intractable due to the exponential nature of the problem. Given a horizon of 10 steps, the number of hypotheses becomes $D_{10} = 2^{10}$, each is a Gaussian conditional distribution. In this and the next experiments the action space is defined as primitive actions, up-down-left-right, in a fixed step size.

The estimated value function obtained from the complete set of hypotheses and the simplified estimator generated using the adaptive pruning approach, as outlined in Section 4.3, are illustrated in figure 4.3. The solver was endowed with an a-priori budget, limiting the maximum loss, denoted as $\epsilon_{\overline{D}}$. Based on the estimator value, the solver determined online which hypotheses to prune and which to retain.

The results indicate that, as the bounds become looser, i.e., when the value of $\epsilon_{\overline{D}}$ increases, the computation time efficiency also increases, trading off efficiency with performance. As the bounds increases beyond the value of 0.7, they become uninformative

since the bounds are larger or smaller than $V_{\max}, V_{\min}$, respectively. On the other hand, when the allowable loss budget was set to zero, no hypotheses were pruned, resulting in identical value estimations for both the pruned and the full estimators, which leads to an identical result as the baseline method of no pruning.

In the second experiment, we aimed to compare the ability of different pruning schemes to complete the task under a limited time-budget of 20 seconds, identical to all solvers. Specifically, we compare the performance of our approach to three types of pruning baselines; no pruning (Full-HB-MCTS), maintaining a fixed number of hypotheses (K-HB-MCTS) and pruning below a threshold value (P_{thresh}-HB-MCTS). Notably, P_{thresh}-HB-MCTS can be seen as an extension of DA-BSP [40], to an MCTS-based algorithm instead of Sparse Sampling, as the earlier is known to perform empirically better. For each pruning method we have experimented with multiple hyperparameters, $\mathrm{P}_{thresh} \in \{0.01, 0.1, 0.3\}$ for P_{thresh}-HB-MCTS, K$\in \{1, 3, 10\}$ for K-HB-MCTS, and $\frac{\epsilon_{\overline{D}}}{V_{\max}} \in \{0.1, 0.2, 0.5\}$ for DA-MCTS. The best are shown in Table 4.1.

In that experiment, the goal of the agent was to reach an ordered set of waypoints, positioned on coordinates $[20, 0], [20, 20], [0, 20]$, see figure 4.4 for an illustration. After performing 60 steps in the environment, the simulation was restarted. The reward was defined as the expected sum of distance to the next waypoint. The state space was defined as the agent pose, and the positions of the landmarks. Ambiguous landmarks were placed in the vicinity of each waypoint to challenge the solvers by causing an exponential increase in the number of hypotheses.

The results of this experiment are presented in Table 4.1. Our findings indicate that the performance of the HB-MCTS algorithm improved when the number of hypotheses was reduced. Given the allocated time budget, maintaining a large set of hypotheses significantly impeded efficiency, leading to a degradation of the planner's exploration. Conversely, maintaining a single hypothesis resulted in an overconfident solver that potentially relied on the wrong association sequence. Our proposed algorithm performed comparably well, as it was able to distinguish between hypotheses with a significant impact on the value function and those with low impact, which can be pruned.

Table 4.1: Reaching waypoints performance over 10 trials. The pruning hyperparameters chosen for the experiments are ($K = 1, P_{thresh} = 0.1, \frac{\epsilon_{\overline{D}}}{V_{\max}} = 0.2$) for K-HB-MCTS, P_{thresh}-HB-MCTS, and DA-MCTS respectively.

Algorithm	Waypoint 1	Waypoint 2	Waypoint 3
DA-MCTS (ours)	**100.0%**	**100.0%**	**90.0%**
Full-HB-MCTS	**100.0%**	30.0%	20.0%
K-HB-MCTS	**100.0%**	80.0%	60.0%
P_{thresh}-HB-MCTS	**100.0%**	80.0%	60.0%

4.5 Conclusions

This chapter proposes a pruning-based approach for efficient autonomous decision-making in environments with ambiguous data associations. The approach models the data association problem as a partially observable Markov decision process (POMDP) and represents multiple data association hypotheses as a belief mixture. The challenge of handling the exponential growth in the number of hypotheses was addressed by pruning the hypotheses while planning, with the number of hypotheses being adapted based on bounds derived on the value function.

The results of our evaluations in simulated environments demonstrate the effectiveness of our approach in handling multi-modal belief hypotheses with ambiguous data associations. Our method provides a practical solution for autonomous agents to make decisions in environments with partial observability and guaranteed performance.

Future research goals include extending the bounds to hybrid belief use-cases, improving solver scalability for ambiguous data associations, efficient recovery of lost hypotheses, and exploring computational burden reduction techniques like merging hypotheses with guarantees.

Chapter 5

Online POMDP Planning with Anytime Deterministic Guarantees

In previous chapters we have considered simplifications of either the state or observation spaces in order to increase planning efficiency, while limiting the loss we incur on the approximated solution. In this chapter, we interleave the approaches and consider simplification of both the state and observation spaces simultaneously. We show that under some restrictions, it is possible to obtain a deterministic bound on the gap between any approximated solution and the optimal one.

Decision-making under incomplete information can be formalized as Partially Observable Markov Decision Processes (POMDPs). Finding an optimal solution to most POMDP problems is computationally intractable, mostly due to a large number of possibilities for the ground truth of the current state, and exponentially increasing possibilities of the future outcomes, commonly referred to as the curse of dimensionality, and the curse of history [45]. As such, most state-of-the-art (SOTA) algorithms aim to find an approximate solution.

One prominent approach to deriving approximate solutions employs an online tree-search paradigm. In this framework, following each real-world decision, an online solver evaluates the current state and projects potential future scenarios. These scenarios are organized within a tree graph structure. As the tree is constructed, the agent assesses the implications of selecting a particular action, subsequently receiving feedback from the environment. This feedback informs the estimation of probabilities for new states, guiding the selection of subsequent actions based on accumulated knowledge. This iterative process continues, building on past outcomes to navigate the decision space.

Given the inherent approximation in these solutions, a natural inquiry regarding the connection between the approximate solution and the actual problem at hand. Some state of the art online algorithms, e.g. [45], offer asymptotic guarantees thus

having no finite time guarantees on the solution quality. A different class of algorithms suggests finite time, but probabilistic guarantees such as [46]. Many algorithms have shown good empirical performance, at the advent of the practical use case of POMDP problems, e.g. [48], but fall short of providing a framework that bridges between the derived solution and the underlying POMDP.

In this work, we focus on deriving deterministic guarantees for POMDPs with discrete state, action and observation spaces. Unlike existing black-box sampling mechanisms employed in algorithms such as [48, 16, 56], our approach assumes access not only to the observation model but also to the transition and the prior models. By leveraging this additional information, we develop novel bounds that necessitate only a subset of the state and observation spaces, enabling the computation of deterministic bounds with respect to the optimal policy at any belief node within the constructed tree. From a practical standpoint, we demonstrate how to harness the theoretical derivations to recent advancements in POMDP approximate solvers, by attaching the bounds to existing state-of-the-art algorithms. We show that despite their stochastic nature, we can guarantee deterministic linkage to the optimal solution with marginal overhead. We extend the approach even further by demonstrating how to utilize the bounds to prune suboptimal branches during exploration, and subsequently select an action based on the deterministic guarantees.

In this chapter, our main contributions are as follows. First, we introduce a simplified POMDP that uses a subset of the state and observation spaces to increase the computational efficiency. Then, we derive deterministic bounds that relate between the former and the non-simplified POMDP. Notably, the bounds are only a function of the states and observations known to the simplified POMDP and hence can be calculated in the planning phase to guide the decision-making and even exploration. We show that utilizing these bounds for exploration results in convergence to the optimal solution of the POMDP in finite time. Based on the derived bounds, we illustrate how to incorporate the bounds into a general structure of common state-of-the-art algorithms. We utilize the bounds for pruning suboptimal actions while exploring the tree via other exploration mechanisms, such as UCT [8]. Last, we demonstrate the practicality of the bounds by experimenting with various algorithms to improve the empirical results of state-of-the-art algorithms in finite-horizon problems.

5.1 Preliminaries

In this chapter, the reward is defined as an expectation over a state-dependent function, $r(b_t, a_t) = \mathbb{E}_{x\sim b_t}[r_x(x, a_t)]$, and is assumed to be bounded by $-\mathcal{R}_{\max} \leq r_x(x, a_t) \leq \mathcal{R}_{\max}$. Consequently, the value function for a policy π over a finite horizon $\mathcal{T}$ is defined as the expected cumulative reward received by executing π and can be computed using

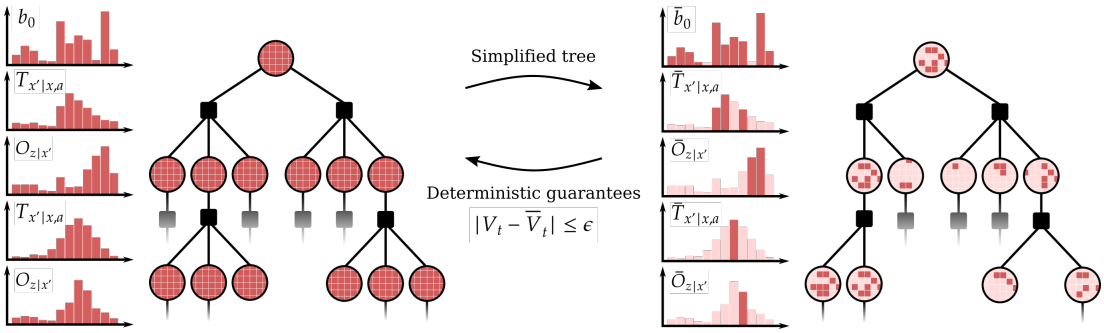


Figure 5.1: The figure depicts two search trees: a complete tree (left) that considers all states and observations at each planning step, and a simplified tree (right) that incorporates only a subset of states and observations, linked to simplified models. Our methodology establishes a deterministic link between these two trees.

the Bellman update equation,

$$V_t^{\pi}(b_t) = r(b_t, \pi_t) + \mathop{\mathbb{E}}_{z_{t+1:\mathcal{T}}} \left[\sum_{\tau=t+1}^{\mathcal{T}} r(b_\tau, \pi_\tau) \right]. \tag{5.1}$$

We use $V_t^{\pi}(b_t)$ and $V_t^{\pi}(H_t)$ interchangeably throughout the chapter. The action-value function is defined by executing action a_t and then following policy π,

$$Q_t^{\pi}(b_t, a_t) = r(b_t, a_t) + \mathop{\mathbb{E}}_{z_{t+1:\mathcal{T}}} \left[\sum_{\tau=t+1}^{\mathcal{T}} r(b_\tau, \pi_\tau) \right]. \tag{5.2}$$

The optimal value function may be computed using Bellman's principle of optimality,

$$V_t^{\pi^*}(b_t) = \max_{a_t} \{ r(b_t, a_t) + \mathop{\mathbb{E}}_{z_{t+1}|a_t,b_t} \left[V_{t+1}^{\pi^*}(b_{t+1}) \right] \}. \tag{5.3}$$

For notational convenience, we introduce a few more simplifying notations; We use $\mathcal{V}_{max,t}, \mathcal{V}_{min,t}$ to denote upper an lower bounds on the value function at time step t. In the simplest case, these may be $\mathcal{V}_{max,t} = (\mathcal{T}-t)\cdot\mathcal{R}_{\max}$, $\mathcal{V}_{min,t} = (t-\mathcal{T})\cdot\mathcal{R}_{\max}$. Additionally, in this chapter we denote a trajectory as, $\tau_t = \{x_0, a_0, z_1, x_1, a_1, \ldots, a_{t-1}, x_t, z_t\}$, and a corresponding probability distribution over the possible trajectories, $\mathbb{P}(\tau_t)$. We denote a policy-dependent trajectory distribution as $\mathbb{P}^{\pi}(\tau_t) \equiv \mathbb{P}(\tau_t \mid b_0, \pi_0, \ldots, \pi_t)$.

5.2 Simplified POMDP

Typically, it is infeasible to fully expand a Partially Observable Markov Decision Process (POMDP) tree due to the extensive computational resources and time required. To address this challenge, we propose two approaches. In the first approach, presented in 5.3.1, we propose a solver that selectively chooses a subset of the observations to branch from, while maintaining a full posterior belief at each node. This allows us to derive an

hypothetical algorithm that directly uses our suggested deterministic bounds to choose which actions to take while exploring the tree. As in most scenarios computing a complete posterior belief may be too expensive, in section 5.3.2 we suggest an improved method that in addition to branching only a subset of the observations, selectively chooses a subset of the states at each encountered belief.

The presented approaches diverge from many existing algorithms that rely on black-box prior, transition, and observation models. Instead, our method directly utilizes state and observation probability values to evaluate both the value function and the associated bounds. In return, an anytime deterministic guarantee on the value function for the derived policy concerning its deviation from the optimal value function is derived.

To that end, we define a simplified POMDP, which is a reduced version of the original POMDP that abstracts or ignores certain states and/or observations. A simplified POMDP, $\bar{M}$, is a tuple $\langle \bar{\mathcal{X}}, \mathcal{A}, \bar{\mathcal{Z}}, \bar{\mathcal{T}}, \bar{O}, \mathcal{R}, \bar{b}_0 \rangle$, where $\bar{\mathcal{X}}, \bar{\mathcal{Z}}, \bar{\mathcal{T}}$ and $\bar{O}$ are the simplified versions of the state and observation spaces, and their corresponding transition and observation models,

$$\bar{b}_0(x) \triangleq \begin{cases} b_0(x) & , \ x \in \bar{\mathcal{X}}_0 \\ 0 & , \ otherwise \end{cases} \tag{5.4}$$

$$\bar{\mathbb{P}}(x_{t+1} \mid x_t, a_t) \triangleq \begin{cases} \mathbb{P}(x_{t+1} \mid x_t, a_t) & , \ x_{t+1} \in \bar{\mathcal{X}}(H_{t+1}^-) \\ 0 & , \ otherwise \end{cases} \tag{5.5}$$

$$\bar{\mathbb{P}}(z_t \mid x_t) \triangleq \begin{cases} \mathbb{P}(z_t \mid x_t) & , \ z_t \in \bar{\mathcal{Z}}(H_t) \\ 0 & , \ otherwise \end{cases} \tag{5.6}$$

where $\bar{\mathcal{X}}(H_{t+1}^-) \subseteq \mathcal{X}$ and $\bar{\mathcal{Z}}(H_t) \subseteq \mathcal{Z}$ may be chosen arbitrarily, e.g. by sampling or choosing a fixed subset a-priori, as the derivations of the bounds are independent of the subset choice. Note that the simplified prior, transition and observation models are unnormalized and do not aim to represent valid distribution functions. For the rest of the sequel we drop the explicit dependence on the history, and denote $\bar{\mathcal{X}}(H_{t+1}^-) \equiv \bar{\mathcal{X}}$, $\bar{\mathcal{Z}}(H_t) \equiv \bar{\mathcal{Z}}$. The action space, $\mathcal{A}$ and prior probability, b_0 are as defined in the original POMDP, M.

With the definition of the simplified POMDP, we define a corresponding simplified

value function,

$$\bar{V}^{\pi}(\bar{b}_0) \triangleq \bar{\mathbb{E}}\left[\sum_{t=0}^{\mathcal{T}} r_x(x_t, a_t)\right] \tag{5.7}$$

$$= \sum_{t=0}^{\mathcal{T}} \sum_{z_{1:t}} \sum_{x_{0:t}} \prod_{k=1}^{t} \bar{\mathbb{P}}(z_k \mid x_k)\bar{\mathbb{P}}(x_k \mid x_{k-1}, \pi_{k-1})\bar{b}(x_0) r_x(x_t, a_t) \tag{5.8}$$

$$= \sum_{t=0}^{\mathcal{T}} \sum_{\tau_t} \bar{\mathbb{P}}^{\pi}(\tau_t) r_x(x_t, a_t), \tag{5.9}$$

where the simplified expectation-like operator, $\bar{\mathbb{E}}[\cdot]$, is taken with respect to the simplified prior, transition and observation models, which do not include the entire distribution, and thus is not a complete expectation.

We use the simplified value function as a computationally-efficient replacement for the theoretical value function; For clarity, the simplified POMDP and consequently all derivations consider a finite-horizon POMDP, but its extension to the discounted infinite horizon case is straightforward, by introducing the discount factor whenever the reward is being used, and an additive term for truncating the tree, $\gamma^t V_{max,t}$, as suggested in, e.g., [27].

In the following sections, we will derive upper and lower bounds between the simplified and the theoretical values of a given policy. Then, we will show how to use the simplification to achieve guarantees with respect to the optimal value function of the original POMDP, and how to utilize these bounds for planning.

5.3 Anytime Deterministic Guarantees for Simplified POMDPs

5.3.1 Simplified Observation Space

We first analyze the performance guarantees of a simplified observation space, while assuming a complete belief update at each considered history node, i.e., $\bar{\mathcal{X}} \equiv \mathcal{X}$. Such an approach is viable when the posterior belief can be calculated efficiently, e.g. when the state spae is sufficiently small. We start by presenting a bound between the simplified value function and the theoretical one of a given policy; then, we provide optimality guarantees for any policy, obtained by solving the simplified POMDP, both in terms of convergence and a deterministic bound, in which the optimal value, for an unknown policy must reside in.

Fixed Policy Guarantees for Simplified Observation Spaces

The following theorem describes the guarantees of the observation-simplified value function with respect to its theoretical value,

Theorem 5.1. *Let b_t belief state at time t, and $\mathcal{T}$ be the last time step of the POMDP. Let $V^{\pi}(b_t)$ be the theoretical value function by following a policy π, and let $\bar{V}^{\pi}(b_t)$ be*

the simplified value function, as defined in (5.7)*, by following the same policy. Then, for any policy* π*, the difference between the theoretical and simplified value functions is bounded as follows,*

$$\left|V^{\pi}(b_t) - \bar{V}^{\pi}(b_t)\right| \leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \overline{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \triangleq \epsilon^{\pi}(b_t). \tag{5.10}$$

Proof. The proof is provided in A.7. ■

Similarly, the action-dependent bound on the value difference, denoted $\epsilon^{\pi}(b_t, a_t)$, is the bound of taking action a_t in belief b_t and following policy π thereafter,

$$\left|Q^{\pi}(b_t, a_t) - \bar{Q}^{\pi}(b_t, a_t)\right| \leq \epsilon^{\pi}(b_t, a_t), \tag{5.11}$$

where,

$$\begin{aligned} \epsilon^{\pi}(b_t, a_t) \triangleq \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} [1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \overline{\mathbb{P}}(z_{t+1} \mid x_{t+1}) \mathbb{P}(x_{t+1} \mid x_t, a_t) \cdot \\ \prod_{k=t+2}^{\tau} \overline{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})]. \end{aligned} \tag{5.12}$$

Importantly, $\epsilon^{\pi}(b_t)$ and $\epsilon^{\pi}(b_t, a_t)$ only contain terms which depend on observations that are within the simplified space, $z \in \bar{\mathcal{Z}}$. This is an essential property of the bounds, as it is a value that can easily be calculated during the planning process and provides a certification of the policy quality at any given node along the tree. Furthermore, it is apparent from (5.10) that as the number of observations included in the simplified set, $\bar{\mathcal{Z}}$, increases, the values of $\epsilon^{\pi}(b_t)$ and $\epsilon^{\pi}(b_t, a_t)$ consequently diminishes,

$$\sum_{z_{1:\tau}} \sum_{x_{0:\tau}} b(x_0) \prod_{k=1}^{\tau} \overline{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \xrightarrow{\bar{\mathcal{Z}} \to \mathcal{Z}} 1$$

leading to a convergence towards the theoretical value function, i.e. $\epsilon^{\pi}(b_t) \to 0$ and $\epsilon^{\pi}(b_t, a_t) \to 0$.

Optimality Guarantees for Simplified Observation Spaces

Theorem 5.1 provides both lower and upper bounds for the theoretical value function, assuming a fixed policy. Using this theorem, we can derive upper and lower bounds for any policy, including the optimal one. This is achieved by applying the Bellman optimality operator to the upper bound in a repeated manner, instead of the estimated value function; In the context of tree search algorithms, our algorithm explores only a subset of the decision tree due to pruned observations. However, at every belief node

encountered during this exploration, all potential actions are expanded. The action-value function of these expanded actions is bounded using the Upper Deterministic Bound, which we now define as

$$\mathrm{UDB}^{\pi}(b_t, a_t) \triangleq \bar{Q}^{\pi}(b_t, a_t) + \epsilon^{\pi}(b_t, a_t) = r(b_t, a_t) + \bar{\mathbb{E}}_{z_{t+1}}[\bar{V}^{\pi}(b_{t+1})] + \epsilon^{\pi}(b_t, a_t). \quad (5.13)$$

In the event that no subsequent observations are chosen for a given history, the value of $\bar{Q}^{\pi}(b_t, a_t)$ simplifies to the immediate reward plus an upper bound for any subsequent policy, given by $\mathcal{R}_{\max} \cdot (\mathcal{T} - t - 1)$. Then, we make the following claim,

Lemma 5.3.1. *The optimal value function can be bounded by,*

$$V^{\pi *}(b_t) \leq \mathrm{UDB}^{\pi^{\dagger}}(b_t), \quad (5.14)$$

where the policy $\pi^{\dagger}$ is determined according to Bellman optimality over the UDB, i.e.

$$\pi^{\dagger}(b_t) = \arg\max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi^{\dagger}}(b_t, a_t) + \epsilon^{\pi^{\dagger}}(b_t, a_t)] = \arg\max_{a_t \in \mathcal{A}} \mathrm{UDB}^{\pi^{\dagger}}(b_t, a_t) \quad (5.15)$$

$$\mathrm{UDB}^{\pi^{\dagger}}(b_t) \triangleq \max_{a_t \in \mathcal{A}} \mathrm{UDB}^{\pi^{\dagger}}(b_t, a_t). \quad (5.16)$$

Proof. The proof is provided in A.6.1. ■

Notably, using UDB to find the optimal policy does not require a recovery of all the observations in the theoretical belief tree, but only a subset which depends on the definition and complexity of the POMDP. Each action-value is bounded by a lower and upper bound, which can be represented as an interval enclosing the theoretical value. When the bound intervals of two candidate actions do not overlap, one can clearly discern which action is suboptimal, rendering its subtree redundant for further exploration. This distinction sets UDB apart from current state-of-the-art online POMDP algorithms. In those methods, any finite-time stopping condition fails to ensure optimality since the bounds used are either heuristic or probabilistic in nature.

In addition to certifying the obtained policy with Bellman optimality criteria, one can utilize UDB as an exploration criteria,

$$a_t = \arg\max_{a_t \in \mathcal{A}} [\mathrm{UDB}^{\pi^{\dagger}}(b_t, a_t)], \quad (5.17)$$

which ensures convergence to the optimal value function, as the number of visited posterior nodes increases.

Corollary 5.2. *By utilizing Lemma 5.3.1 and the exploration criteria defined in* (5.17)*, an increasing number of explored belief nodes guarantees convergence to the optimal value function.*

Proof. The proof is provided in A.8. ■

5.3.2 Simplified State and Observation Spaces

In most scenarios, a complete evaluation of posterior beliefs during the planning stage may pose significant computational challenges. To tackle this issue, we propose the use of a simplified state space in addition to the simplified observation space considered thus far. Specifically, we derive deterministic guarantees of the value function that allow for the selection of a subset from both the states and observations.

We start the analysis of simplifying the state-and-observation spaces by fixing a policy and derive upper and lower bounds for the theoretical, yet unknown, value function at the root node, hereafter referred to as the 'root-value'. This process involves the use of a simplified value function and an additional bonus term, which are easier to compute than the theoretical value function. Considering that various segments of the decision tree contribute differently to the upper bound, we then examine each subtree's contribution separately, which leads to a recursive formulation of the bound. Importantly, these bounds are exclusively derived in relation to, and hold only with respect to, the root node. This is in contrast to the bounds shown in theorem 5.1, which bound the value function of each node in the belief tree.

Using the deterministic bounds at the root allows us to certify the performance of following a particular policy starting from the root of the planning tree. Based on these bounds we extend previous results, shown in theorem 5.3, and show that, (1) exploring the tree with a bound that is formulated with respect to the root node leads to an optimistic estimation of the optimal value function with respect to that root node. (2) Utilizing the bounds for action exploration leads to convergence to the optimal solution of the entire tree. (3) We develop a new method for pruning suboptimal mid-tree action branches. This method includes a bonus term for the upper and lower bounds, accounting for previously unconsidered cumulative probability, enhancing model efficiency by eliminating less optimal paths.

Fixed policy guarantees

We begin by stating the core theorem of our work, which sets forth the upper and lower bounds of a root-value function, with a simplified value function,

Theorem 5.3. *Let b_0 and $\bar{b}_0$ be the theoretical and simplified belief states, respectively, at time $t = 0$, and $\mathcal{T}$ be the last time step of the POMDP. Let $V^\pi(b_0)$ be the theoretical value function by following a policy π, and let $\bar{V}^\pi(\bar{b}_0)$ be the simplified value function by following the same policy, as defined in* (5.7)*. Then, for any policy π, the theoretical value function and at the root is bounded as follows,*

$$\mathcal{L}_0^\pi(H_0) \leq V^\pi(b_0) \leq \mathcal{U}_0^\pi(H_0). \tag{5.18}$$

where,

$$\mathcal{U}_0^\pi(H_0) \equiv \bar{V}^\pi(\bar{b}_0) + \mathcal{V}_{max,0}\left[1 - \sum_{\tau_0}\bar{\mathbb{P}}(\tau_0)\right] + \sum_{t=0}^{\mathcal{T}-1}\mathcal{V}_{max,t+1}\left[\sum_{\tau_t}\bar{\mathbb{P}}^\pi(\tau_t) - \sum_{\tau_{t+1}}\bar{\mathbb{P}}^\pi(\tau_{t+1})\right] \tag{5.19}$$

$$\mathcal{L}_0^\pi(H_0) \equiv \bar{V}^\pi(\bar{b}_0) + \mathcal{V}_{min,0}\left[1 - \sum_{\tau_0}\bar{\mathbb{P}}(\tau_0)\right] + \sum_{t=0}^{\mathcal{T}-1}\mathcal{V}_{min,t+1}\left[\sum_{\tau_t}\bar{\mathbb{P}}^\pi(\tau_t) - \sum_{\tau_{t+1}}\bar{\mathbb{P}}^\pi(\tau_{t+1})\right] \tag{5.20}$$

Proof.. A proof is provided in A.9. ■

A key aspect of Theorem 5.3 is that the bounds it establishes are exclusively dependent on the simplified state and observation spaces. This characteristic is vital in order to compute them during the planning phase.

The intuition behind the result of the derivation can be interpreted as follows; it takes a conservative approach to the value estimation by assuming that every trajectory not observed may obtain an extremum value. Moreover, it allows flexibility in how the trajectories are selected, which are allowed to be chosen arbitrarily in terms of the simplified state space, observation space and the horizon of each trajectory.

The theorem provides bounds for the theoretical value function at the root node of the search tree, given a policy. Using Bellman-like equations, one can restructure the formulation to compute the bounds recursively, which is crucial for making computations in online planning computationally efficient,

$$\mathcal{U}_0^\pi(H_t) \triangleq \sum_{\tau_t\in\mathcal{T}(H_t)}\bar{\mathbb{P}}(\tau_t)r_x(x_t,\pi_t) + \sum_{\tau_t\in\mathcal{T}(H_t)}\bar{\mathbb{P}}(\tau_t)\mathcal{V}_{\max,t} + \sum_{z_{t+1}\in\bar{\mathcal{Z}}(H_t,\pi_t)}\Bigg[\mathcal{U}_0^\pi(H_{t+1}) - \sum_{\tau_{t+1}\in\mathcal{T}(H_{t+1})}\bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\mathrm{m}} \tag{5.21}$$

$$\mathcal{L}_0^\pi(H_t) \triangleq \sum_{\tau_t\in\mathcal{T}(H_t)}\bar{\mathbb{P}}(\tau_t)r_x(x_t,\pi_t) + \sum_{\tau_t\in\mathcal{T}(H_t)}\bar{\mathbb{P}}(\tau_t)\mathcal{V}_{\min,t} + \sum_{z_{t+1}\in\bar{\mathcal{Z}}(H_t,\pi_t)}\Bigg[\mathcal{L}_0^\pi(H_{t+1}) - \sum_{\tau_{t+1}\in\mathcal{T}(H_{t+1})}\bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\mathrm{mi}} \tag{5.22}$$

and,

$$\mathcal{U}_0^\pi(H_T) \triangleq \sum_{\tau_\mathcal{T}\in\mathcal{T}(H_T)}\bar{\mathbb{P}}(\tau_T)r_x(x_T), \qquad \mathcal{L}_0^\pi(H_T) \triangleq \sum_{\tau_\mathcal{T}\in\mathcal{T}(H_T)}\bar{\mathbb{P}}(\tau_T)r_x(x_T). \tag{5.23}$$

where $\mathcal{T}(H_t)$ represent the set of trajectories that consist history H_t, i.e., all trajectories $\mathcal{T}(H_t) = \{(x_{0:t}, a_{0:t-1}, z_{1:t}) \mid (a_{0:t-1}, z_{1:t}) = H_t\}$. The values $\mathcal{U}_0^\pi(H_t)$ and $\mathcal{L}_0^\pi(H_t)$, represent the relative upper and lower bounds of node H_t with respect to the value function at the root, H_0. In other words, they do not represent the bounds of a policy starting from node H_t. The first two summands have a similar structure to the standard

Bellman update operator used in POMDPs, with two main differences. First, the state dependent reward is multiplied by the probability of the entire trajectory from the root node, and not the density value of the belief. Notably, the value of $\sum_{\tau_t \in \mathcal{T}(H_t)} \mathbb{P}(\tau_t)$ will generally not sum to one, due to the dependence of the summed trajectories on the history. Second, there is no expectation operator over the values of the next time step. This is a result of using a distribution over the trajectories, instead of the belief itself. The last summand assigns an optimistic value for the set of trajectories reached to node H_t but not to H_{t+1}.

Optimality Guarantees

We have shown in theorem 5.3 how to calculate bounds for the difference in value functions between the original and the simplified POMDP, given a fixed policy. In this section, we show that by applying Bellman-like optimality operator on $\mathcal{U}_0(H_t)$, the obtained value at the root node is an upper bound for the optimal value function. More formally,

Lemma 5.3.2. *Let $\mathcal{A}$ be the set of actions and $\mathcal{U}_0^\star(H_t)$, $\mathcal{L}_0^\star(H_t)$ be the upper and lower bounds of node H_t chosen according to,*

$$\mathcal{U}_0^\star(H_t) \triangleq \max_{a_t} \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \left[r_x(x_t, a_t) + \mathcal{V}_{\max,t} \right] + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{U}_0^\star(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\max,t} \right] \tag{5.24}$$

$$\mathcal{L}_0^\star(H_t) \triangleq \max_{a_t} \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \left[r_x(x_t, a_t) + \mathcal{V}_{\min,t} \right] + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{L}_0^\star(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\min,t} \right] \tag{5.25}$$

and,

$$\mathcal{U}_0^\star(H_T) \triangleq \sum_{\tau_\mathcal{T} \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r_x(x_T), \qquad \mathcal{L}_0^\star(H_T) \triangleq \sum_{\tau_\mathcal{T} \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r_x(x_T). \tag{5.26}$$

Then, the optimal root-value is bounded by,

$$\mathcal{L}_0^\star(H_0) \leq V^{\pi^*}(H_0) \leq \mathcal{U}_0^\star(H_0). \tag{5.27}$$

The proof is provided in 34.

In this lemma, we establish that employing the 'partial' root-bound is sufficient for ensuring both upper and lower bounds in relation to the optimal value function at the root node. This approach differs from that presented in the previous section (see Lemma 5.3.1). There, each node in the tree was associated with its unique upper bound based on its value function. In contrast, the current lemma demonstrates that

using the 'partial' bound across all nodes in the tree, which is valid only at the root, still guarantees bounded value for the optimal root-value function, while avoiding the requirement to maintain a complete belief at each node of the tree.

Early Stopping Criteria

Lemma 5.3.2 establishes that the recursive Bellman-like optimality operator, can be used to bound the optimal value function at the root. Since the bounds are deterministic, these bounds can be used for eliminating suboptimal actions with full certainty while planning. Then, we define the interval for each action at the root as,

$$I^\star(H_0, a_0) \in [\mathcal{L}_0^\star(H_t, a_0), \mathcal{U}_0^\star(H_0, a_0)]\,, \tag{5.28}$$

and use it as a tool for pruning suboptimal actions once an upper bound of an action falls below the best lower bounds amongst other actions within that node, see figure 5.2 for an illustration.

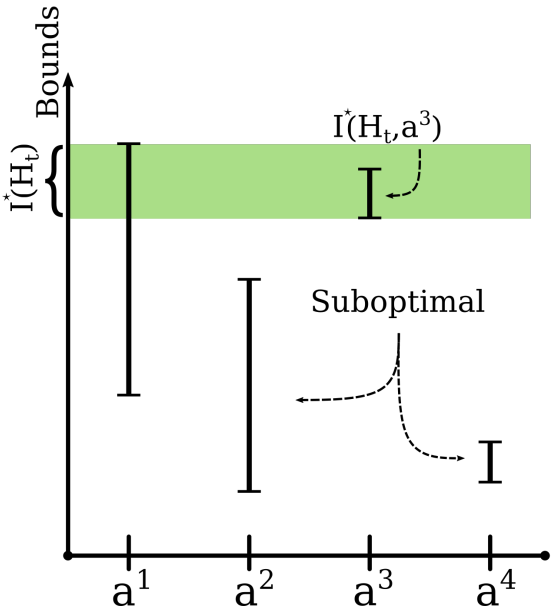


Figure 5.2: Bound intervals for different actions. The optimal value function is guaranteed to be between the maximal lower and upper bounds. As a result, actions a^2 and a^4 are suboptimal and can be pruned safely.

State-of-the-art algorithms such as POMCP and DESPOT employ probabilistic and asymptotic reasoning to approximate the optimal policy, and lack a mechanism to conclusively determine the suboptimality of an action, leading to infinite exploration of suboptimal actions. In contrast, utilizing (5.28) guarantees that once an action is identified suboptimal, it can be safely excluded from further consideration. Since the bounds can be integrated with arbitrary exploration methods, it provides a novel

mechanism for pruning with contemporary SOTA algorithms.

Importantly, this approach introduces a practical stopping criterion for the online tree search process. When the exploration results in only one viable action remaining at the root, it signifies the identification of the optimal action. Note that this does note necessitate exhaustive exploration of the entire tree or complete convergence of the bounds.

Exploration Strategies

One can further utilize the root upper bound to determine the exploration of actions, the simplified state and observation spaces at run time, which guarantees convergences to the optimal value function in finite time, which is novel for online tree search POMDPs solvers to the best of our knowledge. We define the following deterministic exploration strategy,

$$a_t = \arg\max_{a\in\mathcal{A}}\{\sum_{\tau_t\in\mathcal{T}(H_t)}\bar{\mathbb{P}}(\tau_t)r_x(x_t,a) + \sum_{z_{t+1}\in\bar{\mathcal{Z}}(H_t,a)}\mathcal{U}_0^\star(H_{t+1}) + \mathcal{V}_{\max,t}\left[\sum_{\tau_t\in\mathcal{T}(H_t)}\bar{\mathbb{P}}(\tau_t) - \sum_{\tau_{t+1}\in\mathcal{T}(H_t,a_t)}\bar{\mathbb{P}}(\tau_{t+1})\right]\} \tag{5.29}$$

$$z_{t+1} = \arg\max_{o_{t+1}\in\bar{\mathcal{Z}}(H_t,a_t)}\{\mathcal{U}_0^\star((H_t,a_t,o_{t+1})) - \mathcal{L}_0^\star((H_t,a_t,o_{t+1}))\} \tag{5.30}$$

$$x_{t+1} = \arg\max_{x\in\mathcal{X}(H_{t+1})}\{\bar{\mathbb{P}}^\star((\tau_t,a_t,z_{t+1},x)) - \sum_{\tau_\mathcal{T}}\bar{\mathbb{P}}^\star(\tau_T \mid \tau_t,a_t,z_{t+1},x)\}, \tag{5.31}$$

where the actions are chosen by the highest upper bound, sometimes referred to as an *"optimism in face of uncertainty"*, which offers a balance between exploration and exploitation of actions that are possibly optimal or have high uncertainty in their value. Observations are chosen based on the maximum gap between the upper and lower bounds, which results in observations with high uncertainty in their value. Last, we define $\bar{\mathbb{P}}^\star(\tau_t)$ as the probability of a trajectory τ_t under a policy derived from recursive action selection as per (5.29). Subsequently, the selection of states effectively maximizes the difference in probability between the individual trajectory density and the aggregate probability of all sampled trajectories that begin with that particular trajectory.

Lemma 5.3.3. *Performing exploration based on* (5.29)*,* (5.30) *and* (5.31) *ensures that the algorithm converges to the optimal value function within a finite number of planning iterations. The proof is provided in A.6.4.*

Importantly, alternative methods for the state-action-observation exploration are viable and, if given limited planning time, may offer improved performance in practice. Lemma 5.3.3 suggests one way that is guaranteed to converge in finite time. We leave the investigation of other approaches for finite-time convergence using the deterministic bounds for future research.

Moreover, the bounds suggested in this chapter can be integrated with established algorithms like POMCP or DESPOT ([45, 46]), an approach which offers several ad-

vantages over the existing algorithms. First, The quality of their solutions with respect to the optimal value can be assessed and validated. Second, whenever the bounds at the root of the solver do not overlap, the planning session can be terminated early with a guarantee of identifying the optimal action.

5.4 Algorithms

Algorithm 5.1 ALGORITHM-$\mathcal{A}$:

function SEARCH
while time permits **do**
  Generate states x from b_0.
  $\tau_0 \leftarrow x$
  $\bar{\mathbb{P}}_0 \leftarrow b(x = \tau_0 \mid h_0)$
  if $\tau_0 \notin \tau(h_0)$ **then**
    $\bar{\mathbb{P}}(h_0) \leftarrow \bar{\mathbb{P}}(h_0) + \bar{\mathbb{P}}_0$
  end if
  SIMULATE$(h_0, D, \tau_0, \bar{\mathbb{P}}_0)$.
end while
Return

function FWDUPDATE$(ha, haz, \tau_d, \bar{\mathbb{P}}_\tau, x')$
if $\tau_d \notin \tau(ha)$ **then**
  $\tau(ha) \leftarrow \tau(ha) \cup \{\tau_d\}$
  $\bar{R}(ha) \leftarrow \bar{R}(ha) + \bar{\mathbb{P}}_\tau \cdot r_x(x, a)$
end if
$\tau_d \leftarrow \tau_d \cup \{x'\}$
$\bar{\mathbb{P}}_\tau \leftarrow \bar{\mathbb{P}}_\tau \cdot Z_{z|x'} \cdot T_{x'|x,a}$
if $\tau_d \notin \tau(haz)$ **then**
  $\bar{\mathbb{P}}(haz) \leftarrow \bar{\mathbb{P}}(haz) + \bar{\mathbb{P}}_\tau$
  $\tau(haz) \leftarrow \tau(haz) \cup \{\tau_d\}$
end if
Return

function SIMULATE$(h, d, \tau_d, \bar{\mathbb{P}}_d)$
if $d = 0$ **then**
  Return
end if
Select action a.
Generate next states and observations, x', z.
$\tau_d,\ \bar{\mathbb{P}}_\tau \leftarrow$FWDUPDATE$(ha, haz, \tau_d, \bar{\mathbb{P}}_\tau, x')$
Select next observation z.
SIMULATE$(haz, d-1, \tau_d, \bar{\mathbb{P}}_\tau)$
BWDUPDATE(h, ha, d)
Return

function BWDUPDATE(h, ha, d)
$\epsilon(ha) = \gamma^{D-d} V_{\max,d}(\bar{\mathbb{P}}(h) - \bar{\mathbb{P}}(ha)) + \gamma^{D-d-1} \cdot V_{\max,d+1}(\bar{\mathbb{P}}(ha) - \sum_{z|ha} \bar{\mathbb{P}}(haz))$
$U(ha) = \bar{R}(ha) + \gamma \sum_{z|ha} U(haz) + \epsilon(ha)$
$L(ha) = \bar{R}(ha) + \gamma \sum_{z|ha} L(haz) - \epsilon(ha)$
$U(h) \leftarrow \max_{a'}\{U(ha')\}$
$L(h) \leftarrow \max_{a'}\{L(ha')\}$
Return

In this section we aim to describe how to fit our bounds to a blueprint of a general algorithm, named ALGORITHM $-\ \mathcal{A}$, which serves as an abstraction to many existing algorithms. Then, we explicitly describe two algorithms, DB-POMCP, an adaptation to POMCP that uses UCB for exploration, and our deterministic bounds for decision-making, and RB-POMCP, a particle-based solver that utilizes the bounds both for decision-making and exploration.

To compute the deterministic bounds, we utilize Bellman's update and optimality criteria. This approach naturally fits dynamic programming approaches such as DESPOT [57] and AdaOPS [56]. However, it may also be attached with algorithms that rely on Monte-Carlo estimation, such as POMCP [45], by viewing the search tree as a policy tree.

While the analysis presented in section 5.3 is general and independent of the selection mechanism of the states or observations, we focus on sampling as a way to choose the simplified states at each belief node and the observations to branch from. Furthermore, the selection of the subspaces $\bar{\mathcal{X}}, \bar{\mathcal{Z}}$ need not be fixed, and may change over the course of time, similar to state-of-the-art algorithms, such as [16, 45, 46, 48, 56]. Alternative selection methods may also be feasible, as sampling from the correct distribution is not required for the bounds to hold. Importantly, attaching our bounds to arbitrary exploration mechanism certifies the algorithm solution with deterministic bounds to the optimal solution, and may result in an improved decision making, as will be shown in the experimental section.

ALGORITHM $-\ \mathcal{A}$ is outlined in algorithm 5.1. For clarity of exposition, we assume the following; at each iteration a single state particle is propagated from the root node to the leaf (line 2 of function SEARCH). The selection of the next state and observations are done by sampling from the observation and transition models (line 5), and each iteration ends with the full horizon of the POMDP (lines 2). However, none of these are a restriction of our approach and may be replaced with arbitrary number of particles, arbitrary state and observation selection mechanism and a single or multiple expansions of new belief nodes at each iteration.

To compute the UDB value, we require both the state trajectory, denoted as τ, and its probability value, $\mathbb{P}_\tau$. We use the state trajectory as a mechanism to avoid duplicate summation of an already accounted for probability value and is utilized to ascertain its uniqueness at a belief node. The probability value, $\mathbb{P}_\tau$, is the likelihood of visiting a trajectory $\tau = \{x_0, a_0, x_1, z_1, \ldots, a_{t-1}, x_t, z_t\}$ and is calculated as the product of the prior, transition and observation likelihoods (line 6). If a trajectory was not previously observed in a belief node, its reward value is multiplied by the likelihood of the trajectory. Each trajectory likelihood is maintained as part of a cumulative sum of all visited trajectories in the node. This cumulative sum is then used to calculate the upper and lower bounds, which are shown in lines 1-2. The term computed in line 1 represents the loss of holding only a subset of the states in node ha from the set in node h, plus the loss of having only a partial set of posterior nodes and a subset of their states. $V_{\max,d}$ represents an upper bound for the value function. A simple bound on the value function can be $V_{\max,d} = \mathcal{R}_{\max} \cdot (D-d)$, but other more sophisticated bounds may also be used. In the experimental section we show that despite the additional overhead, utilizing the deterministic bounds, (5.20) and (5.19), within the actual decision-making improves the results of the respective algorithms.

5.4.1 DB-POMCP

DB-POMCP uses theorem 5.3 for decision-making once an optimal action was found or at time-out given limited planning time. In aligning Algorithm 5.1 with the POMCP framework, the action exploration process determined by the Upper Confidence Bounds

for Trees (UCT) criterion,

$$UCT(H_t, a_t) = \hat{Q}^{mean}(H_t, a_t) + c\sqrt{\frac{log(N(H_t))}{N(H_t, a_t)}}, \tag{5.32}$$

where $\hat{Q}^{mean}$ is the average of the cumulative sums obtained from sampled explorations, and c is a tunable constant that trades-off exploration and exploitation during planning. Following this criterion, each state and observation is then sampled according to their respective transition and observation models. The original POMCP method, as discussed in [45], employs Monte-Carlo rollouts for value estimation and refrains from adding new nodes during these rollouts. During our evaluations we saw a negligible difference in performance, thus we avoid presenting rollouts to algorithm 5.1 for simplicity. However, DB-POMCP supports both settings.

5.4.2 RB-POMCP

Root-Bounded POMCP (RB-POMCP) differs from DB-POMCP in that it uses a different exploration method. We denote it RB-POMCP to emphasize that the bounds hold only in the root node, and are not valid for any node along the tree, yet unlike DB-POMCP the bounds are used for exploration in any part of the tree. The RB-POMCP methodology draws inspiration from the Monte-Carlo approach suggested the original POMCP algorithm and innovates by incorporating upper and lower bounds, as defined in (5.24) and (5.25), to guide both the exploration and the decision-making processes.

The RB-POMCP framework is constructed based on the structure outlined in Algorithm 5.1, which necessitates specific implementations for abstract state, action, and observation exploration functions. In our approach, we opt for an approximation to the exploration mechanism proposed in section 5.3.2. More precisely, while we adhere to the action exploration strategy described in the lemma, we simplify the observation and state exploration components by employing basic Monte-Carlo sampling techniques, akin to those used in the standard POMCP algorithm. This modification is intended to enhance the algorithm's planning efficiency without compromising the integrity of the algorithm bounds. The remainder of the RB-POMCP algorithm adheres closely to the procedures specified in Algorithm 5.1. Additionally, we use pruning and stopping criteria, as described in 5.3.2.

5.4.3 Time complexity

The time complexity for each posterior node, primarily depends on the specific algorithm being used. In the case of dynamic programming methods, such as DESPOT and AdaOPS, there is a negligible added computational complexity detailed below. In the case of Monte Carlo methods, such as POMCP, the computational complexity is $O(|\mathcal{A}|)$ attributed mainly to the action-selection, while our approach adds another

linear time complexity term, making it $O(|\mathcal{A}| + |\bar{\mathcal{Z}}|)$ due to the summation over the simplified observation space. During each iteration of the algorithm, an "IF" statement is used to determine whether a specific trajectory has already been encountered at the current node. This verification process can potentially result in an added linear complexity of $O(D)$, where D represents the planning horizon. However, this overhead can be circumvented by assigning a unique ID value to each trajectory at the previous step and subsequently checking whether a pair, comprising the ID value and the new state, has already been visited. This approach reduces the overhead to an average time complexity of $O(1)$ by utilizing hash maps efficiently.

5.5 Experiments

Our primary contribution is of a theoretical nature, yet we conducted experiments to evaluate the practical applicability of our proposed methodologies. Initially, we adopted a hybrid strategy, such as DB-POMCP, by incorporating our deterministic bounds exclusively for the decision-making, while relying on existing exploration strategies such as POMCP and DESPOT. Essentially, this approach enhances the POMCP and DESPOT frameworks by equipping them with mechanisms that ensure bounded sub-optimality. In a subsequent experimental setup, we applied the deterministic bounds to both the exploration and decision-making phases, based on the methodologies outlined in section 5.4.2. We then compared the empirical performance of using the deterministic bounds solely for decision-making to the baseline algorithms without the incorporation of any deterministic bounds. Our findings indicate that while the application of deterministic bounds to decision-making can enhance performance, this strategy becomes less effective in identifying the optimal action as the complexity of the POMDP increases. Conversely, when the deterministic bounds are applied to both exploration and decision-making (section 5.4.2), the results demonstrate a linear increase in planning time proportional to the size of the POMDP, indicating better scalability.

5.5.1 Deterministic-Bounds for Decision-Making

In this subsection, we focus on the application of deterministic bounds exclusively for decision-making. This approach involves using a predefined exploration strategy during the planning phase, but making the final action selection based on the deterministic bounds as shown in (5.24). The comparative results for the standard and deterministically-bounded versions of the POMCP and DESPOT algorithms are presented in Table 5.1. These versions, labeled DB-POMCP and DB-DESPOT, adhere to the original exploration criteria of their respective algorithms but select actions based on the highest lower bound, as specified in (5.20).

Our experimental analysis reveals that, in addition to offering a level of optimality certification for the chosen actions, utilizing deterministic bounds for action selection

can enhance the expected cumulative reward. It is important to note, however, that this method does not always lead to better outcomes. Specifically, it may not be advantageous in situations where the highest lower bound is less than other available upper bounds (for instance, comparing actions a^1 and a^3 in figure 5.2). This limitation is evident in the results for the Laser Tag POMDP, a considerably larger problem compared to the other POMDPs evaluated, where the deterministic bounds did not yield performance improvements.

Table 5.1: Performance comparison with and without deterministic bounds, for short horizon, $H = 5$.

Algorithm	Tiger POMDP	Laser Tag	Discrete Light Dark	Baby POMDP
DB-DESPOT (ours)	3.7 ±0.48	−5.3 ±0.14	−5.3 ±0.01	−3.9 ±0.56
AR-DESPOT	2.8 ±0.55	−5.1 ±0.14	−61.5 ±5.80	−5.4 ±0.85
DB-POMCP (ours)	3.0 ±0.21	−4.0 ±0.24	−3.7 ±0.82	−4.5 ±0.57
POMCP	2.2 ±0.76	−3.9 ±0.27	−4.5 ±1.15	−5.4 ±0.63

5.5.2 Root-Bounds for Decision-Making and Exploration

Table 5.2: Performance comparison with and without deterministic bounds, for medium horizon, $H = 15$.

Algorithm	Tiger POMDP	Rock Sample	Navigate to Goal	Baby POMDP
RB-POMCP (ours)	1.5 ±0.76	8.5 ±0.22	61.2 ±0.71	−12.0 ±0.27
DB-POMCP (ours)	−1.1 ±0.15	7.9 ±0.21	62.4 ±0.75	0.0 ±0.00
POMCP	−5.6 ±0.24	5.7 ±0.20	68.5 ±0.69	−12.5 ±0.27

The performance outcomes presented in Table 5.2 reveal that the RB-POMCP algorithm typically matches or surpasses the standard POMCP in various tested environments, except for the Navigate to Goal POMDP scenario. The limited performance in this particular context can be attributed to the nature of RB-POMCP's exploration strategy, which is designed to assure optimality over extended planning periods but does not inherently guarantee enhanced results within limited planning durations. Unlike probabilistic algorithms that leverage statistical concentration inequalities—such as the Hoeffding inequality employed in the Upper Confidence Bounds for Trees (UCT) [27] exploration mechanism of POMCP—RB-POMCP adopts a more cautious strategy. This approach entails considering both worst-case and best-case scenarios to establish a deterministic link with the optimal value may not always translate to superior immediate performance due to its conservative nature.

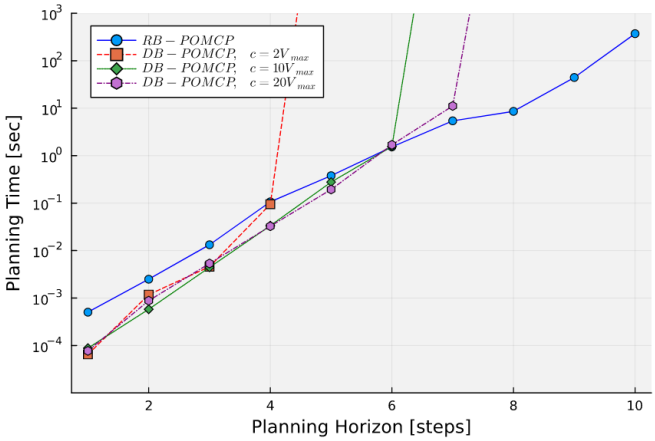


Figure 5.3: The graphs show the measured planning time for RB-POMCP and DB-POMCP to find the optimal action under different UCT coefficient values. All simulation runs were capped at 3,600 seconds.

5.5.3 Planning for optimal action

To highlight the differences between RB-POMCP and DB-POMCP, we examined each algorithm's planning time to deterministically identify the optimal value, as depicted in Figure 5.3. Notably, conventional state-of-the-art algorithms, such as POMCP and DESPOT, cannot deterministically identify the optimal action within a finite timeframe and are thus not considered in this analysis.

DB-POMCP incorporates the Upper Confidence Bounds for Trees (UCT) method for exploration. However, its exploration strategy lacks awareness of the deterministic bounds of the optimal value function, leading to insufficient guidance toward actions that may be optimal. Despite significantly increasing the exploration coefficient beyond the values suggested in previous works [45, 48], our findings, as presented in Figure 5.3, demonstrate that the exploration bonus diminishes too rapidly, effectively limiting further exploration of potentially optimal actions. While UCT, in theory, explores the belief tree indefinitely, in practical scenarios, the exploration rate of new branches diminishes exponentially over time, making it less effective in environments where identifying the optimal action in a reasonable time is crucial. Conversely, RB-POMCP directly utilizes upper and lower bounds information, facilitating a more targeted search for the optimal value. This approach leads to a planning duration that scales linearly with the problem size, as evidenced in Figure 5.3, highlighting its efficiency in identifying optimal actions within a finite timeframe.

5.5.4 Technical Details

The implementation of our algorithm written in the Julia programming language, using the Julia POMDPs package for evaluation and the vanilla POMDP versions, provided by [13]. This package primarily supports infinite horizon problems; however, we modified it to also handle finite-horizon POMDPs. The experiments were conducted on a

computing platform consisting of an Intel(R) Core(TM) i7-7700 processor with 8 CPUs operating at 3.60GHz and 15.6 GHz. The hyper-parameters for the POMCP and AR-DESPOT solvers, and further details about the POMDPs used for our experiments are detailed in appendix A.5.

5.6 Conclusions

In this work, we presented a novel methodology aimed at offering anytime, deterministic guarantees for approximate POMDP solvers. These solvers strategically leverage a subset of the state and observation spaces to alleviate the computational overhead. Our key proposition elucidates a linkage between the optimal value function, which is inherently computationally intensive, and a more tractable approximation frequently employed in contemporary algorithms. In the first part of the chapter, we derived the theoretical relationship between the use of a selective subset of states and observations in a planning tree. One contribution of this work is an extension of previously published result on upper deterministic bound (UDB) to govern exploration in the case of simplified state and observation spaces, while being theoretically guaranteed to converge to the optimal value. This approach, however, may be computationally infeasible in many practical POMDPs, due to its need to iterate through all states and observations at each node. Thus, we provide two novel algorithms, DB-POMCP and RB-POMCP, that approximate this approach, while still being able to provide a deterministic relationship to the optimal value and provide a stopping criteria for when the planning has converged to its optimal value. Additionally, we provide a method to attach our bounds to existing state-of-the-art algorithms. We extend this approach by providing the ability to prune sub-optimal branches within the exploration phase. We have outlined how our methodology can be integrated within these algorithms. Finally, to illustrate the practical utility of our derivations, we evaluate DB-POMCP and RB-POMCP against state-of-the-art algorithms and highlight the differences in performance to find the optimal action between the two.

Appendix A

Appendices

A.1 Adaptive Information Belief Space Planning

A.1.1 Proofs

Lemma 1

The proof is provided for continuous state space; The discrete case obtained similarly by changing integrals to summations.

Proof.

$$\sum_{n=1}^{N_z} \bar{\mathbb{P}}\left(z^n \mid H^-\right) \mathcal{H}(\bar{b}) =$$
$$-\sum_{n=1}^{N_z} \bar{\mathbb{P}}\left(z^n \mid H^-\right) \int_x \bar{\mathbb{P}}(x \mid H) \cdot log(\bar{\mathbb{P}}(x \mid H)) \tag{A.1}$$

applying Bayes' rule for $\bar{\mathbb{P}}(x \mid H)$,

$$-\sum_{n=1}^{N_z} \int_x \bar{O}(z^n \mid x)\mathbb{P}\left(x \mid H^-\right) \tag{A.2}$$
$$\cdot log\left(\frac{\bar{O}(z^n \mid x)\mathbb{P}\left(x \mid H^-\right)}{\int_{x'} \bar{O}(z^n \mid x')\mathbb{P}\left(x' \mid H^-\right)}\right)$$

Splitting summation to follow the partitioning of the abstract observation model,

$$-\sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} \int_x \bar{O}(z^k \mid x)\mathbb{P}\left(x \mid H^-\right) \tag{A.3}$$
$$\cdot log\left(\frac{\bar{O}(z^k \mid x)\mathbb{P}\left(x \mid H^-\right)}{\int_{x'} \bar{O}(z^k \mid x')\mathbb{P}\left(x' \mid H^-\right)}\right)$$

By construction, $\bar{O}(z \mid x)$ has uniform distribution for z^k, where $k \in [K(c-1)+1, Kc]$. Thus,

$$-\sum_{c=1}^{C}\left[\sum_{k=K(c-1)+1}^{Kc} 1\right]\int_x \bar{O}(z^{Kc} \mid x)\mathbb{P}\left(x \mid H^-\right) \tag{A.4}$$

$$\cdot log\left(\frac{\bar{O}(z^{Kc} \mid x)\mathbb{P}\left(x \mid H^-\right)}{\int_{x'}\bar{O}(z^{Kc} \mid x')\mathbb{P}\left(x' \mid H^-\right)}\right) =$$

$$-\sum_{c=1}^{C} K \cdot \int_x \bar{O}(z^{Kc} \mid x)\mathbb{P}\left(x \mid H^-\right) \tag{A.5}$$

$$\cdot log\left(\frac{\bar{O}(z^{Kc} \mid x)\mathbb{P}\left(x \mid H^-\right)}{\int_{x'}\bar{O}(z^{Kc} \mid x')\mathbb{P}\left(x' \mid H^-\right)}\right) =$$

$$K \cdot \sum_{c=1}^{C}\bar{\mathbb{P}}\left(z^{Kc} \mid H^-\right)\mathcal{H}(\bar{b})$$

which concludes the proof. ■

Lemma 2

Proof. We begin with,

$$\bar{\mathbb{E}}_o \left[\mathbb{E}_{x \sim \bar{b}} \left[r_x \left(x, a \right) \right] \right] \tag{A.6}$$

by definition of $\bar{\mathbb{E}}_o \left[\cdot \right]$, $\bar{b}(x)$,

$$\sum_{n=1}^{N_z} \bar{\mathbb{P}} \left(z^n \mid H^- \right) \left[\sum_{x \in S} \bar{\mathbb{P}}(x \mid z^n, H^-) r_x \left(x, a \right) \right] \tag{A.7}$$

applying chain rule,

$$\begin{aligned} &\sum_{n=1}^{N_z} \sum_{x \in S} \bar{\mathbb{P}}(x, z^n \mid H^-) r_x \left(x, a \right) = \\ &\sum_{n=1}^{N_z} \sum_{x \in S} \bar{O}(z^n \mid x) b^-(x) r_x \left(x, a \right) \end{aligned} \tag{A.8}$$

we split the sum over the observations to comply with the abstraction partitioning and use the the abstract observation model definition, (4),

$$\sum_{x \in S} \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} \frac{\sum_{m=K(c-1)+1}^{Kc} O(z^m \mid x)}{K} b^-(x) r_x \left(x, a \right) \tag{A.9}$$

we then arrive at the desired result,

$$\sum_{x \in S} \sum_{n=1}^{N_z} O(z^n \mid x) b^-(x) r_x \left(x, a \right) = \tag{A.10}$$

$$\mathbb{E}_o \left[\mathbb{E}_{x \sim b} \left[r_x \left(x, a \right) \right] \right] \tag{A.11}$$

Theorem 1

Proof. For clarity, we omit the time index in the derivation, the result holds for any time step. We use H^- to denote past history while excluding last observation. We also use b and $\mathbb{P}(x \mid z, H^-)$ interchangeably. Rearranging the abstraction from (1),

$$\sum_{k=1}^{K} \bar{O}\left(z^k \mid x\right) = K \cdot \bar{O}(z^b \mid x) \doteq \sum_{k=1}^{K} O\left(z^k \mid x\right) \qquad \forall b \in [1, K]$$

Plugging it to the expected entropy term, (11),

$$\bar{\mathbb{E}}_z\left[\mathcal{H}\left(\bar{b}\right)\right] - \mathbb{E}_z\left[\mathcal{H}\left(b\right)\right] = \tag{A.12}$$

$$\sum_{i=1}^{N_o} \bar{\mathbb{P}}\left(z_i \mid H^-\right) \mathcal{H}\left(\bar{b}\right) - \sum_{i=1}^{N_o} \mathbb{P}\left(z_i \mid H^-\right) \mathcal{H}\left(b\right) \tag{A.13}$$

expanding the entropy term,

$$-\sum_{i=1}^{N_o} \bar{\mathbb{P}}\left(z_i \mid H^-\right) \int_x \bar{\mathbb{P}}\left(x \mid z_i, H^-\right) log\left(\bar{b}\right) \tag{A.14}$$

$$+\sum_{i=1}^{N_o} \mathbb{P}\left(z_i \mid H^-\right) \int_x \mathbb{P}\left(x \mid z_i, H^-\right) log\left(b\right)$$

by Bayes' rule,

$$-\sum_{i=1}^{N_o} \int_x \bar{O}\left(z_i \mid x\right) \mathbb{P}\left(x \mid H^-\right) log\left(\bar{b}\right) \tag{A.15}$$

$$+\sum_{i=1}^{N_o} \int_x O\left(z_i \mid x\right) \mathbb{P}\left(x \mid H^-\right) log\left(b\right)$$

a change in the order of summation and integral and a split of $N_o = C \cdot K$ result in,

$$-\int_x \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} \bar{O}\left(z_k \mid x\right) \mathbb{P}\left(x \mid H^-\right) log\left(\bar{b}\right) \tag{A.16}$$

$$+\int_x \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} O\left(z_k \mid x\right) \mathbb{P}\left(x \mid H^-\right) log\left(b\right)$$

By plugging-in the definition of the abstract model,

$$
\begin{aligned}
&-\int_x \sum_{c=1}^{C} \left[\sum_{k=K(c-1)+1}^{Kc} \frac{\sum_{\bar{k}=K(c-1)+1}^{Kc} O\left(z_{\bar{k}} \mid x\right)}{K} \right] && \text{(A.17)}\\
&\mathbb{P}\left(x \mid H^{-}\right) log\left(\bar{b}\right)\\
&+\int_x \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} O\left(z_k \mid x\right) \mathbb{P}\left(x \mid H^{-}\right) log\left(b\right) =\\
&-\int_x \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} O\left(z_k \mid x\right) \mathbb{P}\left(x \mid H^{-}\right) log\left(\bar{b}\right) && \text{(A.18)}\\
&+\int_x \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} O\left(z_k \mid x\right) \mathbb{P}\left(x \mid H^{-}\right) log\left(b\right) =\\
&\sum_{i=1}^{N_o} \mathbb{P}\left(z_i \mid H^{-}\right) \int_x b \cdot log\left(\frac{b}{\bar{b}}\right) = && \text{(A.19)}\\
&E_z\left[\mathcal{D}_{KL}\left(b||\bar{b}\right)\right] \geq 0
\end{aligned}
$$

(A.19) obtained by applying similar steps in reverse order. The last equality holds since KL-divergence is non-negative and so is its expectation. It is left to prove the upper bound; Applying Bayes rule to the nominator and denominator of (A.19),

$$
\begin{aligned}
&\sum_{i=1}^{N_o} \mathbb{P}\left(z_i \mid H^{-}\right) \int_x blog\left(\frac{O\left(z_i \mid x\right)}{\bar{O}\left(z_i \mid x\right)}\right) && \text{(A.20)}\\
&+\sum_{i=1}^{N_o} \mathbb{P}\left(z_i \mid H^{-}\right) log\left(\frac{\bar{\mathbb{P}}\left(z_i \mid H^{-}\right)}{\mathbb{P}\left(z_i \mid H^{-}\right)}\right) \int_x bds
\end{aligned}
$$

By construction of the abstract observation model,

$$
\begin{aligned}
&\sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} \mathbb{P}\left(z_k \mid H^{-}\right) \int_x b \cdot log\left(\frac{O\left(z_k \mid x\right) \cdot K}{\sum_{k=K(c-1)+1}^{Kc} O\left(z_{\bar{k}} \mid x\right)}\right) ds\\
&+\sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} \mathbb{P}\left(z_k \mid H^{-}\right) log\left(\frac{\bar{\mathbb{P}}\left(z_k \mid H^{-}\right)}{\mathbb{P}\left(z_k \mid H^{-}\right)}\right)\\
&\leq log(K) \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{Kc} \mathbb{P}\left(z_k \mid H^{-}\right) \int_x bds + 0 = log(K).
\end{aligned}
$$

The inequality is due to positiveness of the denominator in the first term and Jensen'x inequality in the second term. we end up with,

$$
0 \leq \bar{\mathbb{E}}_z\left[\mathcal{H}\left(\bar{b}\right)\right] - \mathbb{E}_z\left[\mathcal{H}\left(b\right)\right] \leq log(K). \tag{A.21}
$$

Corollary 1.1

Proof. From Lemma 2 it is clear that the expected state-dependent reward is unaffected by the abstraction, and thus will not affect the value function. For the sake of conciseness and clarity, we prove the case that the value function depends only on the entropy. The general case derived similarly by applying Lemma 2 instead of the expected state-dependent reward.

$$V^{\pi}(b_t) = \sum_{n=1}^{N_z} \mathbb{P}\left(z_{t+1}^{n} \mid H_{t+1}^{-}\right) \left[-\mathcal{H}(b_{t+1}) + V^{\pi}(b_{t+1})\right]$$

expanding the value function,

$$\begin{aligned} &- \sum_{n=1}^{N_z} \mathbb{P}\left(z_{t+1}^{n} \mid H_{t+1}^{-}\right) [\mathcal{H}(b_{t+1}) \\ &+ \sum_{n'=1}^{N_z} \mathbb{P}\left(z_{t+2}^{n'} \mid H_{t+2}^{-}\right) \mathcal{H}(b_{t+2}) + \cdots]] \end{aligned} \tag{A.22}$$

by linearity of expectation,

$$- \mathbb{E}_{z_{t+1}}\left[\mathcal{H}(b_{t+1})\right] + \mathbb{E}_{z_{t+1}}\left[\mathbb{E}_{z_{t+2}}\Big[\mathcal{H}(b_{t+2})\Big]\right] + \cdots \tag{A.23}$$

using Theorem 1 for each of the expected entropy terms separately until time-step $\mathcal{T} - 1$,

$$\begin{aligned} V^{\pi}(b_t) \geq &- \Big[\bar{\mathbb{E}}[\mathcal{H}(\bar{b}_{t+1})] + log(K)\Big] \\ &- \mathbb{E}_{z_{t+1}}\Big[\bar{\mathbb{E}}_{z_{t+2}}[\mathcal{H}(\bar{b}_{t+2})] + log(K)\Big] \cdots \end{aligned} \tag{A.24}$$

$$\begin{aligned} = &- \bar{\mathbb{E}}_{z_{t+1}}\Big[\mathcal{H}(\bar{b}_{t+1})\Big] \\ &- \mathbb{E}_{z_{t+1}}\bar{\mathbb{E}}_{z_{t+2}}\Big[\mathcal{H}(\bar{b}_{t+2})\Big] \cdots + \mathcal{T} \cdot log(K) \end{aligned} \tag{A.25}$$

applying similar steps in reverse order yields the abstract value function,

$$\bar{V}^{\pi}(b_t) + \mathcal{T} \cdot log(K)$$

$$\implies \bar{V}^{\pi}(b_t) - V^{\pi}(b_t) \leq \mathcal{T} \cdot log(K).$$

Following the same derivation and applying the other side of the inequality of Theorem 1, completes the derivation for the entropy as reward. Using the more general reward, (1), and applying Lemma 2, yields the proof for corollary 1.1,

$$0 \leq \bar{V}^{\pi}(b_t) - V^{\pi}(b_t) \leq \mathcal{T} \cdot \omega_2 log(K).$$

Expected Entropy Estimation

We derive an estimator to the expected differential entropy with continuous observation space. The discrete state or observation spaces follows similar derivation by replacing integrals with summations.

$$\mathbb{E}[\mathcal{H}(b_t)] = -\int_{z_t} p(z_t \mid H_t^-) \int_{x_t} p(x_t \mid z_t, H_t^-) \cdot log(p(x_t \mid z_t, H_t^-)) \tag{A.26}$$

applying Bayes' rule,

$$\begin{aligned}\mathbb{E}[\mathcal{H}(b_t)] = &-\int_{z_t}\int_{x_t} p(x_t, z_t \mid H_t^-) \\ &\cdot log\left(O(z_t \mid x_t)\int_{x_{t-1}} T(x_t \mid x_{t-1}, a_{t-1}) b(x_{t-1})\right) \\ &+\int_{z_t}\int_{x_t} p(x_t, z_t \mid H_t^-) \cdot log(p(z_t \mid H_t^-))\end{aligned} \tag{A.27}$$

by chain rule and marginalization,

$$\begin{aligned}\mathbb{E}[\mathcal{H}(b_t)] = &-\int_{z_t}\int_{x_t} O(z_t \mid x_t) b^-(x_t) \\ &\cdot log\left(O(z_t \mid x_t)\int_{x_{t-1}} T(x_t \mid x_{t-1}, a_{t-1}) b(x_{t-1})\right) \\ &+\int_{z_t}\int_{x_t} O(z_t \mid x_t) b^-(x_t) \\ &\cdot log\left(\int_{x_t} O(z_t \mid x_t) b^-(x_t)\right)\end{aligned} \tag{A.28}$$

using particle filter, the belief represented as a set of weighted particles, $\{(x^1, q^1), \ldots, (x^i, q^i), \ldots, (x^n, q^n$
Where q^i denotes the weight of particle i.

$$\begin{aligned}\mathbb{E}[\mathcal{H}(b_t)] \approx &-\int_{z_t} \eta_t \sum_{i=1}^{n} O\left(z_t \mid x_t^i\right) q_{t-1}^i \cdot \\ &\cdot log\left(O\left(z_t \mid x_t^i\right) \sum_{j=1}^{n} p\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j\right) \\ &+\int_{z_t} \eta_t \sum_{i=1}^{n} O\left(z_t \mid x_t^i\right) q_{t-1}^i \cdot \\ &\cdot log\left(\sum_i O\left(z_t \mid x_t^i\right) q_{t-1}^i\right)\end{aligned} \tag{A.29}$$

where $\eta_t = \int_{z_t} \sum_{i=1}^{n} O\left(z_t \mid x_t^i\right) q_{t-1}^i$ normalizes the estimator for the probability function so that it sums to 1. Then, we approximate expectation over the observation space using observation samples, and query the likelihood model conditioned on the state samples,

$O\left(z^m \mid x^i\right) \quad \forall z^m \in \{z^1, \ldots, z^M\},$

$$\hat{\mathbb{E}}[\mathcal{H}(\hat{b}_t)] = -\overline{\eta}_t \sum_{m=1}^{M} \sum_{i=1}^{n} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i \cdot \tag{A.30}$$

$$\cdot log\left(O\left(z_t^m \mid x_t^i\right) \sum_{j=1}^{n} T\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j\right)$$

$$+\overline{\eta}_t \sum_{m=1}^{M} \left[\sum_{i=1}^{n} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i\right]$$

$$\cdot log\left(\sum_{i'=1}^{n} O\left(z_t^m \mid x_t^{i'}\right) q_{t-1}^{i'}\right)$$

$$\overline{\eta}_t = \frac{1}{\sum_{m=1}^{M} \sum_{i=1}^{n} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i} \tag{A.31}$$

which concludes the derivation.

Theorem 2

Note that the reward function, (1), is built of two terms, state dependent reward and entropy,

$$R(b,a,b') = \omega_1 \mathbb{E}_{x\sim b'}\left[r_x(x,a)\right] + \omega_2 \mathcal{H}(b'). \tag{A.32}$$

For clarity, we divide the proof into two parts,

$$\hat{\bar{\mathbb{E}}}_z\left[R\left(\hat{b},a,\hat{\bar{b}}'\right)\right] - \hat{\mathbb{E}}_z\left[R\left(\hat{b},a,\hat{b}'\right)\right] = \tag{A.33}$$

$$\omega_1\left(\hat{\bar{\mathbb{E}}}_z\left[\mathbb{E}_{x\sim\hat{\bar{b}}'}\left[r_x(x,a)\right]\right] - \hat{\mathbb{E}}_z\left[\mathbb{E}_{x\sim\hat{b}'}\left[r_x(x,a)\right]\right]\right) \tag{A.34}$$

$$+\omega_2\left(\hat{\bar{\mathbb{E}}}_z\left[\mathcal{H}\left(\hat{\bar{b}}\right)\right] - \hat{\mathbb{E}}_z\left[\mathcal{H}\left(\hat{b}\right)\right]\right).$$

The first is about the difference in expected entropy, which is similar in spirit to the proof of Theorem 1. The second follows the claim and proof of Lemma (2). We begin with the difference of the expected entropy. For clarity, we derive the upper and lower bounds separately. For the upper bound,

Proof. In the following we directly plug-in the expected entropy estimator, with both the abstract observation model and the original observation model. For clarity, we split the expression into two parts and deal with each separately.

$$\begin{aligned}
&\hat{\bar{\mathbb{E}}}_z\left[\mathcal{H}\left(\hat{\bar{b}}\right)\right] - \hat{\mathbb{E}}_z\left[\mathcal{H}\left(\hat{b}\right)\right] \\
&= \underbrace{-\overline{\eta}_t \sum_{m=1}^{M}\sum_{i=1}^{n} \overline{O}\left(z_t^m \mid x_t^i\right) q_{t-1}^i}_{(a)} \\
&\quad \underbrace{\cdot log\left(\overline{O}\left(z_t^m \mid x_t^i\right)\sum_{j=1}^{n} T\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j\right)}_{(a)} \\
&\quad \underbrace{+\overline{\eta}_t \sum_{m=1}^{M}\sum_{i=1}^{n} \overline{O}\left(z_t^m \mid x_t^i\right) q_{t-1}^i \cdot log\left(\sum_{i=1}^{n} \overline{O}\left(z_t^m \mid x_t^i\right) q_{t-1}^i\right)}_{(b)} \\
&\quad \underbrace{+\overline{\eta}_t \sum_{m=1}^{M}\sum_{i=1}^{n} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i}_{(a)} \\
&\quad \underbrace{\cdot log\left(O\left(z_t^m \mid x_t^i\right)\sum_{j=1}^{n} T\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j\right)}_{(a)} \\
&\quad \underbrace{-\overline{\eta}_t \sum_{m=1}^{M}\sum_{i=1}^{n} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i \cdot log\left(\sum_{i=1}^{n} O\left(z_t^m \mid x_t^i\right) q_{t-1}^i\right)}_{(b)}
\end{aligned} \tag{A.35}$$

In the first expression we start by splitting the summation to sum over its clusters and sum over the components of each cluster,

$$(a) = \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \tag{A.36}$$

$$\cdot log\left(\frac{O\left(z_t^k \mid x_t^i\right) \sum_{j=1}^{n} p\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j}{\overline{O}\left(z_t^k \mid x_t^i\right) \sum_{j=1}^{n} p\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j}\right)$$

$$(a) = \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \tag{A.37}$$

$$\cdot log\left(\frac{O\left(z_t^k \mid x_t^i\right)}{\overline{O}\left(z_t^k \mid x_t^i\right)}\right)$$

using the abstract model, (4),

$$(a) = \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \tag{A.38}$$

$$\cdot log\left(\frac{K \cdot O\left(z_t^k \mid x_t^i\right)}{\sum_{k=K(c-1)+1}^{K\cdot c} O\left(z_t^k \mid x_t^i\right)}\right). \tag{A.39}$$

since the denominator within the log is a sum of positive values, the following clearly holds,

$$(a) \leq \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \cdot log(K) \tag{A.40}$$

by taking the constant $log(K)$ out of the summation, the rest sums to one, so $(a) \leq log(K)$. Next we bound the second expression from above,

$$(b) = \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \tag{A.41}$$

$$\cdot log\left(\frac{\sum_{i=1}^{n} \overline{O}\left(z_t^k \mid x_t^i\right) q_{t-1}^i / \overline{\eta}_t}{\sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i / \overline{\eta}_t}\right)$$

applying Jensen'x inequality,

$$(b) \leq log\left(\overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} \overline{O}\left(z_t^k \mid x_t^i\right) q_{t-1}^i\right) \tag{A.42}$$

by recalling the definition of the normalizer, we end up with $log(1) = 0$ ■

Last, we provide a proof for the lower bound,

Proof.

$$\hat{\bar{\mathbb{E}}}_z\left[\mathcal{H}\left(\hat{\bar{b}}\right)\right] - \hat{\mathbb{E}}_z\left[\mathcal{H}\left(\hat{b}\right)\right] = \\ -\overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \\ \cdot log\Big[\frac{\overline{O}\left(z_t^k \mid x_t^i\right) \sum_{j=1}^{n} T\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j}{\sum_{i=1}^{n} \overline{O}\left(z_t^k \mid x_t^i\right) q_{t-1}^i} \\ \cdot \frac{\sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i}{O\left(z_t^k \mid x_t^i\right) \sum_{j=1}^{n} T\left(x_t^i \mid x_{t-1}^j, a_{t-1}\right) q_{t-1}^j}\Big] \tag{A.43}$$

since $log(x) \leq x - 1,\ \forall x > 0$,

$$\hat{\bar{\mathbb{E}}}_z\left[\mathcal{H}\left(\hat{\bar{b}}\right)\right] - \hat{\mathbb{E}}_z\left[\mathcal{H}\left(\hat{b}\right)\right] \geq \\ \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \\ \cdot \left(1 - \frac{\overline{O}\left(z_t^k \mid x_t^i\right)}{\sum_{i=1}^{n} \overline{O}\left(z_t^k \mid x_t^i\right) q_{t-1}^i} \cdot \frac{\sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i}{O\left(z_t^k \mid x_t^i\right)}\right) \tag{A.44}$$

rearranging terms,

$$1 - \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i \tag{A.45}$$

$$\cdot \frac{\overline{O}\left(z_t^k \mid x_t^i\right)}{\sum_{i=1}^{n} \overline{O}\left(z_t^k \mid x_t^i\right) q_{t-1}^i} \cdot \frac{\sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i}{O\left(z_t^k \mid x_t^i\right)} \tag{A.46}$$

we conclude with,

$$1 - \overline{\eta}_t \sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K\cdot c} \sum_{i=1}^{n} O\left(z_t^k \mid x_t^i\right) q_{t-1}^i = 0 \tag{A.47}$$

We now derive the second part of Theorem 2, i.e. for the difference of expected state-dependent reward.

Lemma A.1.1. *The value of the estimated expected state-dependent reward is not affected by the abstraction shown in (4), i.e.,*

$$\hat{\bar{\mathbb{E}}}_z\left[\mathbb{E}_{x\sim\hat{\bar{b}}'}\left[r_x(x,a)\right]\right] = \hat{\mathbb{E}}_z\left[\mathbb{E}_{x\sim\hat{b}'}\left[r_x(x,a)\right]\right] \tag{A.48}$$

Proof.

$$\hat{\bar{\mathbb{E}}}_z \left[\mathbb{E}_{x \sim \hat{\bar{b}}_t'} \left[r_x(x_t, a_t) \right] \right] = \tag{A.49}$$

$$\sum_{m=1}^{M} \bar{\mathbb{P}}(z_t^m \mid H_t^-) \sum_{i=1}^{n} \bar{\mathbb{P}}(x_t^i \mid z_t^m, H_t^-) r_x(x_t^i, a_t) \tag{A.50}$$

applying chain rule,

$$\sum_{m=1}^{M} \sum_{i=1}^{n} \bar{\mathbb{P}}(x_t^i, z_t^m \mid H_t^-) r_x(x_t^i, a_t), \tag{A.51}$$

then applying chain-rule from the other direction and using the markovian assumption of the observation model,

$$\sum_{m=1}^{M} \sum_{i=1}^{n} \bar{O}(z_t^m \mid x_t^i) b_t^- \cdot r_x(x_t^i, a_t). \tag{A.52}$$

Applying the transition function on particles from b_{t-1}, does not alter their weights, therefore we receive the following expression,

$$\sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K \cdot c} \sum_{i=1}^{n} \bar{O}\left(z_t^k \mid x_t^i\right) q_{t-1}^i r_x(x_t^i, a_t) \tag{A.53}$$

Using (1),

$$\sum_{c=1}^{C} \sum_{k=K(c-1)+1}^{K \cdot c} \sum_{i=1}^{n} \frac{\sum_{\bar{k}=K(c-1)+1}^{K \cdot c} O\left(z_t^{\bar{k}} \mid x_t^i\right)}{K} q_{t-1}^i r_x(x_t^i, a_t), \tag{A.54}$$

followed by canceling the summation over k with K in the denominator,

$$\sum_{m=1}^{M} \sum_{i=1}^{n} O(z_t^m \mid x_t^i) b_t^- \cdot r_x(x_t^i, a_t). \tag{A.55}$$

We then end up with the desired result,

$$\hat{\bar{\mathbb{E}}}_z \left[\mathbb{E}_{x \sim \hat{\bar{b}}'} \left[r_x(x, a) \right] \right] = \hat{\mathbb{E}}_z \left[\mathbb{E}_{x \sim \hat{b}'} \left[r_x(x, a) \right] \right] \tag{A.56}$$

To conclude the proofs of Theorem 2, note that,

$$0 \leq \omega_1 \left(\hat{\bar{\mathbb{E}}}_z \left[\mathbb{E}_{x \sim \hat{\bar{b}}'} \left[r_x(x, a) \right] \right] - \hat{\mathbb{E}}_z \left[\mathbb{E}_{x \sim \hat{b}'} \left[r_x(x, a) \right] \right] \right) \tag{A.57}$$

$$+ \omega_2 \left(\hat{\bar{\mathbb{E}}}_z \left[\mathcal{H}\left(\hat{\bar{b}} \right) \right] - \hat{\mathbb{E}}_z \left[\mathcal{H}\left(\hat{b} \right) \right] \right) \leq \omega_2 log(K) \tag{A.58}$$

Corollary 2.1

The proof of 2.1 follows closely to the proof in A.4. Replacing the exact value function with its estimated counterpart from A.6 yields the desired result.

A.2 AI-FSSS

In this section we present the main procedures to derive our algorithm. The variables used in Algorithm A.1 are b, ba and b' which represent a belief node, a predicted belief node, i.e. after performing an action and posterior belief, after incorporating a measurement. $C(\cdot)$ denotes a list of their corresponding children. a and $z_{\{1,\dots,K\}}$ denotes an action and a list of K sampled observations respectively. $\bar{P}_{z|x}$ is a list holding the abstract probability values of the measurement model, as in equation (4). $R_{state}(\cdot,\cdot)$ denotes a state-dependent reward function, which may be defined arbitrarily. γ denotes the discount factor. LB, UB and N are all initialized to zero. ROLLOUT performs a predefined policy. In our experiments, we chose uniform distribution over all actions for the rollout policy. Algorithm A.2 uses b_{init}, which represents the initial belief at the root node, n is the number of iterations and d_{max}, the maximum depth of the planning tree.

Algorithm A.1 AI-FSSS

Procedure: SIMULATE(b,d)

if d $= 0$ **then**
  Return 0, 0
else if $|C(b)| < |\mathcal{A}|$ **then**
  $a, z_{\{1,\dots,K\}} \leftarrow$GEN$(b, K)$
  $\bar{P}_{z|x} \leftarrow$ ABSTRACTOBS$(ba, z_{\{1,\dots,K\}})$ // eq.(4)
  $\bar{\mathbb{E}}[\mathcal{R}(ba)] \leftarrow$EXPECTEDREWARD$(b, a, ba, \bar{P}_{z|x})$
else
  $a \leftarrow$ SELECTACTION(b)
end if
$lb \leftarrow \bar{\mathbb{E}}[\mathcal{R}(ba)]$
$ub \leftarrow \bar{\mathbb{E}}[\mathcal{R}(ba)] + log(K)$
if $0 < N(ba) < K$ **then**
  $z \leftarrow$ POP $(z_{\{1,\dots,K\}})$
  $b' \leftarrow$ POSTERIOR(b, a, z)
  $V_{LB}, V_{UB} \leftarrow$SIMULATE$(b', d-1)$
else if $N(ba) = K$ **then**
  $b' \leftarrow \arg\min_{b'} N(b')$
  $V_{LB}, V_{UB} \leftarrow$SIMULATE$(b', d-1)$
else if $N(ba) = 0$ **then**
  $V_{LB}, V_{UB} \leftarrow$ROLLOUT$(ba, d-1)$
end if
$LB(ba) \leftarrow lb + \frac{\gamma V_{LB} + (|C(ba)|-1)(LB(ba)-lb)}{|C(ba)|}$
$UB(ba) \leftarrow ub + \frac{\gamma V_{UB} + (|C(ba)|-1)(LB(ba)-ub)}{|C(ba)|}$
$LB(b) \leftarrow \max_a LB(ba)$
$UB(b) \leftarrow \max_a UB(ba)$
$N(b) \leftarrow N(b) + 1$
$N(ba) \leftarrow N(ba) + 1$
Return $LB(b), UB(b)$

Algorithm A.2 SOLVE

Procedure: SOLVE

1: **for** $i \in 1 : n$ **do**
2: SIMULATE(b_{init}, d_{max})
3: **end for**
4: *action* $\leftarrow$ ADAPTBOUNDS(b_{init})
5: **return** *action*

Algorithm A.3 REFINE

Procedure: REFINE(b,ba,d)

if IsLeaf(b) **then**
  Return 0, 0
else if ABSTRACT(ba) **then**
  $r_{old} \leftarrow$REUSEREWARD(ba)
  $P_{z|x} \leftarrow$ ORIGINALOBSMODEL($ba, z_{\{1,\dots,K\}}$)
  $\mathbb{E}[\mathcal{H}(ba)] \leftarrow$EXPECTEDENTROPY($b, ba, P_{z|x}$)
  $r \leftarrow r_{old} + \omega_2(\mathbb{E}[\mathcal{H}(ba)] - \bar{\mathbb{E}}[\mathcal{H}(ba)])$
else
  $r \leftarrow r_{old}$
end if
$b' \leftarrow \arg\max_{b'}(UB(b') - LB(b'))$
$a' \leftarrow \arg\max_{a'}(UB(b'a') - LB(b'a'))$
$V_{LB}, V_{UB} \leftarrow$REFINE($b', b'a', d-1$)
$LB(ba) \leftarrow lb + \frac{\gamma V_{LB} + (|C(ba)|-1)(LB(ba)-lb)}{|C(ba)|}$
$UB(ba) \leftarrow ub + \frac{\gamma V_{UB} + (|C(ba)|-1)(LB(ba)-ub)}{|C(ba)|}$
$LB(b) \leftarrow \max_a LB(ba)$
$UB(b) \leftarrow \max_a UB(ba)$
return $LB(b), UB(b)$

Procedure: ADAPTBOUNDS(b_{init})

while $\max_{a^+ \in \mathcal{A}} LB(b_{init}a^+) < \max_{a \in \mathcal{A} \setminus a^+} UB(b_{init}a)$ **do**
  $a^* \leftarrow \arg\max_{a \in \mathcal{A}} LB(b_{init}a)$
  REFINE($b_{init}, b_{init}a^*, d$)
end while
Return a^*

A.2.1 Implementation Details

Domain

We compared the different algorithms on a two-dimensional Light Dark environment. In this domain, the unobserved state of the agent is its pose, (X, Y), defined relative to a global coordinate frame, located at $(0, 0)$. There are 9 possible actions, eight of which has one unit of translation, and they differ from each other by the direction which is equally spaced on a circle, the ninth action has zero translation. We denote the transition model as $x' = f(x, a, w)$. At each time step, the agent receives a noisy estimate of its position as an observation, denoted by $z = h(x, v)$. In our experiments we chose w and v to be distributed according to a Gaussian noise, although in general they may be arbitrary. The reward function defined as the negative weighted sum of distance to goal and entropy,

$$r_x(b, a) = -\mathrm{E}_b[\|x - x_g\|] - \mathcal{H}(b), \tag{A.59}$$

The prior belief assumed to be Gaussian, $b_0 = \mathcal{N}([0,0], \Sigma_0)$. In all our experiments, we employ a receding horizon approach. At each iteration we calculate a solution from scratch and share no information across different time steps.

Domain - Total Return Evaluation

We performed the experiments on a modification of Light Dark 2D and added forbidden regions to the environment. Whenever the agent crosses to a forbidden region, a -10 reward was added to its immediate reward. Also, we added +10 reward whenever the agent reached the goal and stayed there until the episode terminated.

Hyperparameters

Here we present the hyperparameters used to evaluate the total return performance.

AI-FSSS			
n	C	K	
20	4	4	
FFFS			
n	C		
20	4		
PFT-DPW			
n	c[1]	k_o	α_o
20	1	4	0.014

Table A.1: Hyperparameters used in the experiments.

[2] c controls the bonus of the UCB function, which is different from the observation branching factor in FSSS and AI-FSSS, C.

A.3 Monte Carlo Planning in Hybrid Belief POMDPs

A.3.1 Theoretical analysis

Lemma A.3.1. *HB-MCP state-dependent reward estimator,* $\hat{\mathcal{R}}_X \triangleq \frac{1}{N}\sum_{i,j=1}^{N} \lambda_t^{i,j} \frac{1}{n_X}\sum_{k=1}^{n_X} r_x(X_t^{i,j,k}, a_t)$*, is unbiased.*

Proof.. If states are sampled i.i.d. for each hypothesis, then the expected value of the reward estimator, $\hat{\mathcal{R}}_X$, is,

$$
\begin{aligned}
\mathbb{E}[\hat{\mathcal{R}}] &= \int \mathbb{Q}(\hat{\mathcal{R}}_X \mid H_t)\hat{\mathcal{R}}_X d\hat{\mathcal{R}}_X \\
&= \int\int\int \mathbb{Q}(\hat{\mathcal{R}}_X, b, x_{1:n} \mid H_t)\hat{\mathcal{R}}_X dx_{1:n} db d\hat{\mathcal{R}}_X \\
&= \int\int\int \mathbb{Q}(\hat{\mathcal{R}}_X \mid x_{1:n})\mathbb{Q}(b, x_{1:n} \mid H_t)\hat{\mathcal{R}}_X dx_{1:n} db d\hat{\mathcal{R}}_X \\
&= \int\int\int \mathbb{Q}(\hat{\mathcal{R}}_X \mid x_{1:n})\mathbb{Q}(x_{1:n} \mid b, H_t)\mathbb{Q}(b \mid H_t)\hat{\mathcal{R}}_X dx_{1:n} db d\hat{\mathcal{R}}_X \\
&= \int\int \mathbb{Q}(\hat{\mathcal{R}}_X \mid x_{1:n})\mathbb{Q}(x_{1:n} \mid b_t, H_t)\hat{\mathcal{R}}_X dx_{1:n} d\hat{\mathcal{R}}_X \\
&= \int\int \mathbb{Q}(\hat{\mathcal{R}}_X \mid x_{1:n}) \left[\sum_{i,j} \mathbb{Q}(x_{1:n} \mid b_t, \beta_{0:t}^{i,j}, H_t)\mathbb{Q}(\beta_{0:t}^{i,j} \mid b_t, H_t)\right] \hat{\mathcal{R}}_X dx_{1:n} d\hat{\mathcal{R}}_X \\
&= \int\int \mathbb{Q}(\hat{\mathcal{R}}_X \mid x_{1:n}) \left[\sum_{i,j} \mathbb{Q}(x_{1:n} \mid b_t, \beta_{0:t}^{i,j}, H_t)\mathbb{Q}(\beta_{0:t}^{i,j} \mid H_t)\right] \hat{\mathcal{R}}_X dx_{1:n} d\hat{\mathcal{R}}_X \\
&= \int \hat{\mathcal{R}}_X(x_{1:n}) \left[\sum_{i,j} \mathbb{Q}(x_{1:n} \mid b_t, \beta_{0:t}^{i,j}, H_t)\mathbb{Q}(\beta_{0:t}^{i,j} \mid H_t)\right] dx_{1:n} \\
&= \sum_{i,j} \mathbb{Q}(\beta_{0:t}^{i,j} \mid H_t) \int \mathbb{Q}(x_{1:n} \mid b_t, \beta_{0:t}^{i,j}, H_t)\hat{\mathcal{R}}_X(x_{1:n}) dx_{1:n} \\
&= \sum_{i,j} \mathbb{Q}(\beta_{0:t}^{i,j} \mid H_t) \int \mathbb{Q}(x_{1:n} \mid \beta_{0:t}^{i,j}, H_t)\hat{\mathcal{R}}_X(x_{1:n}) dx_{1:n} \\
&= \mathbb{E}_{\mathbb{Q}}\mathbb{E}_{b[X_t]_{\beta_{0:t}}} \hat{\mathcal{R}}_X(x_{1:n}) = \mathbb{E}\left[\frac{1}{N}\sum_{i,j=1}^{N} \lambda_t^{i,j} \frac{1}{n_X}\sum_{k=1}^{n_X} r_x(X_t^{i,j,k}, a_t)\right] \\
&= \mathbb{E}_{\mathbb{Q}}\left[\frac{1}{N}\sum_{i,j=1}^{N} \lambda_t^{i,j} \mathbb{E}_{b[X_t]_{\beta_{0:t}}^{i,j}}\left[\frac{1}{n_X}\sum_{k=1}^{n_X} r_x(X_t^{i,j,k}, a_t)\right]\right] \\
&= \frac{1}{N}\sum_{i,j=1}^{N} \mathbb{E}_{\mathbb{Q}}\left[\frac{\mathbb{P}}{\mathbb{Q}}\frac{1}{n_X}\sum_{k=1}^{n_X} \mathbb{E}_{b[X_t]_{\beta_{0:t}}}\left[r_x(X_t^{i,j,k}, a_t)\right]\right] \\
&= \mathbb{E}_{\mathbb{P}}\left[\mathbb{E}_{b[X_t]_{\beta_{0:t}}} r_x(X_t, a_t)\right] \triangleq \mathcal{R}_X
\end{aligned}
$$

where $\mathbb{P} = \mathbb{P}(\beta_{0:t} \mid H_t)$, $\mathbb{Q} = \mathbb{Q}(\beta_{0:t} \mid H_t)$, and N and n_X denote the number of samples from $\mathbb{Q}$ and $b[X_t]_{\beta_{0:t}}^{i,j}$ respectively. ■

Lemma A.3.2. *Given an unbiased reward estimator, $\hat{\mathcal{R}}$, the value-function estimator used in HB-MCP is unbiased.*

Proof. First, note that the value function of time step $t+1$ can be written as,

$$\begin{aligned}\mathbb{E}_{z_{t+1:\tau}}[\sum_{\tau=t+1}^{\mathcal{T}}\mathcal{R}_\tau] &= \mathbb{E}_{z_{t+1}}\left[\mathcal{R}_{t+1}+\mathbb{E}_{z_{t+2:\tau}}\left[V_{t+2}^\pi\right]\right] \\ &= \underbrace{\mathbb{E}_{\beta_{0:t}}\mathbb{E}_{\beta_{t+1}|\beta_{0:t}}\mathbb{E}_{z_{t+1}|\beta_{0:t+1}}\left[\mathcal{R}_{t+1}\right]}_{\triangleq\alpha_{t+1}}+\mathbb{E}\left[V_{t+2}^\pi\right].\end{aligned} \tag{A.60}$$

and its corresponding estimator,

$$\hat{\alpha}_{t+1} \triangleq \hat{\mathbb{E}}_{\mathbb{Q}}\left[\frac{\mathbb{P}\left(\beta_{t+1}^i \mid \beta_{0:t}^j, H_{t+1}^-\right)}{\mathbb{Q}\left(\beta_{t+1}^i \mid \beta_{0:t}^j, H_0\right)}\lambda_t^j\hat{\mathbb{E}}_{z_{t+1}|\beta_{0:t+1},H_{t+1}^-}[\hat{\mathcal{R}}_{t+1}]\right]. \tag{A.61}$$

Then,

$$\begin{aligned}\mathbb{E}[\hat{\alpha}_{t+1}] &= \mathbb{E}\left[\hat{\mathbb{E}}_{\beta_{0:t+1}^{i,j}|H_{t+1}^-\sim\mathbb{Q}}\left[\frac{\mathbb{P}\left(\beta_{t+1}^i \mid \beta_{0:t}^j, H_{t+1}^-\right)}{\mathbb{Q}\left(\beta_{t+1}^i \mid \beta_{0:t}^j, H_0\right)}\lambda_t^j\cdot\hat{\mathbb{E}}_{z_{t+1}|\beta_{0:t+1},H_{t+1}^-}[\hat{\mathcal{R}}_{t+1}]\right]\right] \\ &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N\frac{\mathbb{P}\left(\beta_{t+1}^i \mid \beta_{0:t}^j, H_{t+1}^-\right)}{\mathbb{Q}\left(\beta_{t+1}^i \mid \beta_{0:t}^j, H_0\right)}\frac{\mathbb{P}\left(\beta_{0:t}^j \mid H_t\right)}{\mathbb{Q}\left(\beta_{0:t}^j \mid H_0\right)}\cdot\frac{1}{n_z}\sum_{k=1}^{n_z}\hat{\mathcal{R}}_{t+1}\right] \\ &= \frac{1}{N}\sum_{i=1}^N\mathbb{E}_{\mathbb{Q}}\left[\frac{\mathbb{P}\left(\beta_{0:t+1}^{i,j} \mid H_{t+1}^-\right)}{\mathbb{Q}\left(\beta_{0:t+1}^{i,j} \mid H_0\right)}\frac{1}{n_z}\sum_{k=1}^{n_z}\mathbb{E}_z\mathbb{E}_{\mathcal{R}}[\hat{\mathcal{R}}_{t+1}]\right] \\ &= \mathbb{E}_{\beta_{0:t+1}|H_{t+1}^-\sim\mathbb{P}}[\mathbb{E}_{z_{t+1}|\beta_{0:t+1},H_{t+1}^-}[\mathcal{R}_{t+1}]] = \mathbb{E}_{z_{t+1}|H_{t+1}^-}[\mathcal{R}_{t+1}]\end{aligned}$$

Continuing recursively on the value function yields the desired result. ■

A.3.2 Implementation details - vanilla-HB-MCTS

Algorithms A.4 and A.5 describe the main procedures of vanilla-HB-MCTS. Algorithm A.4 follows PFT-DPW [48] closely. Line 3 in Algorithm A.4 performs action selection based on the UCT exploration bonus. In our experimental setting, we assumed discrete action space, and thus avoided action progressive widening, which can otherwise be replaced with Line 3. Line 4 performs observation progressive widening, which resamples previously seen observations. This step is required to avoid shallow trees due to a continuous observation space, see [48] for further details. Algorithm A.5 computes the pruned-posterior belief, given the multi-hypotheses posterior belief from the previous time-step and the selected action.

Algorithm A.4 vanilla-HB-MCTS

Procedure:SIMULATE(b, h, d)
if d $= 0$ **then**
  Return 0
end if
$a \leftarrow \arg\max_{\bar{a}} Q(b\bar{a}) + c\sqrt{\frac{log(N(b))}{N(b\bar{a})}}$
if $|C(ba)| \leq k_o N(ba)^{\alpha_o}$ **then**
  $b' \leftarrow$ PRUNEDPOSTERIOR(b, a)
  $r \leftarrow$ REWARD(b, a)
  $C(ba) \cup \{(b', r)\}$
  $R \leftarrow r+$ROLLOUT$(b', d-1)$
else
  $b', r \leftarrow$ Sample uniformly from $C(ba)$
  $R \leftarrow r+$SIMULATE$(b', d-1)$
end if
$N(b) \leftarrow N(b) + 1$
$N(ba) \leftarrow N(ba) + 1$
$Q(ba) \leftarrow Q(ba) + \frac{R-Q(ba)}{N(ba)}$
return R

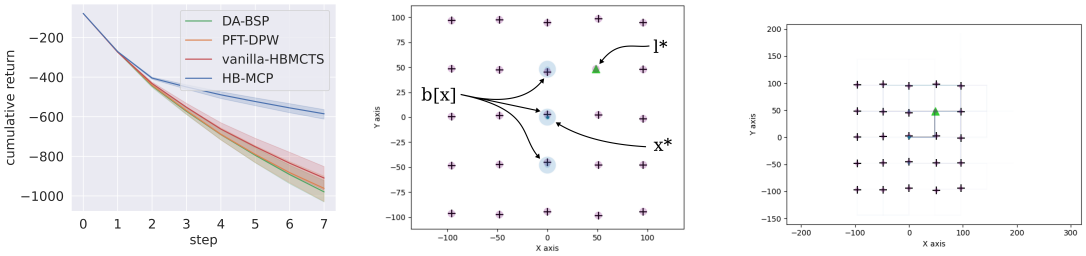
Algorithm A.5 PrunedPosterior

Procedure:PRUNEDPOSTERIOR(b, a)
// $b \triangleq \{b_t^j, \omega_t^j\}_{j=1}^M$
$z \leftarrow$ SAMPLEOBSERVATION(b, a)
$\{\omega_{t+1}^{i,j}\}_{i=1,j=1}^{L,M} \leftarrow$ COMPUTEWEIGHTS(b, a, z) //*eq.*(3.4)
$\{\omega_{t+1}^{i,j}\}_{i=1,j=1}^{L^s(j),M} \leftarrow$ PRUNE$(\{\omega_{t+1}^{i,j}\}_{i=1,j=1}^{L,M})$
$\{\bar{\omega}_{t+1}^{i,j}\}_{i=1,j=1}^{L^s(j),M} \leftarrow$ NORMALIZE$(\{\omega_{t+1}^{i,j}\}_{i=1,j=1}^{L^s(j),M})$
for $j \in [1, M]$ **do**
  for $i \in [1, L^s(j)]$ **do**
    $b_{t+1}^{i,j} \leftarrow \Psi(b_t^j, a, z, i)$ // *eq.* (3.5)
    b'.append$(\{b_{t+1}^{i,j}, \bar{\omega}_{t+1}^{i,j}\})$
  end for
end for
Return b'

A.3.3 Results

This subsection is intended to provide more information about the experiments that appear in the chapter. Specifically, we provide the trajectories performed by HB-MCP and attempt to interpret the results below. In table A.2 we provide the hyperparameters used in our experiments and in table A.3 we provide a numeric values for the average cumulative reward of our experiments.

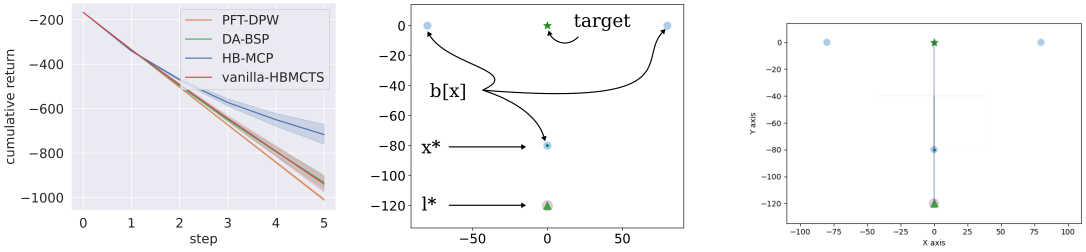
Aliased matrix. There are many ambiguous, evenly spaced landmarks around the agent, along with its ambiguous initial pose, as shown in figure A.1b. The intuitive way to reduce the uncertainty of the belief would be to first disprove wrong hypotheses, and then pass near as many landmarks as possible, such that they would be within the sensing range. The easiest way to disambiguate hypotheses would be to use the unique landmark (see figure A.1b). It is clearly shown in figure A.1c that the agent indeed prioritizes the unique landmark before passing near landmarks. Note that the unique landmark would only be visible (and thus provide observation) if the ground-truth position of the landmark is within the sensing range of the ground-truth pose of the agent. It can also be seen from figure A.1a that after two macro-steps, which is the



(a) cumulative return (b) initial belief (c) trajectories

Figure A.1: The goal of the agent is to minimize the uncertainty of its pose and the location of all landmarks. (a) Mean and standard deviation of the cumulative reward, over 100 trials (higher is better). (b) Illustration of the initial belief of the agent. x^* denotes the ground truth pose of the agent. l^* denotes a unique landmark. The agent receives as a prior three hypotheses at different locations, drawn as blue ellipses. (c) Ground-truth trajectories are visualized in transparent color, illustrated on top of the initial belief, such that multiple similar trajectories appear in a moreopaque color.

distance from the unique landmark, the descent in cumulative reward becomes less steep, and significantly outperform other algorithms.

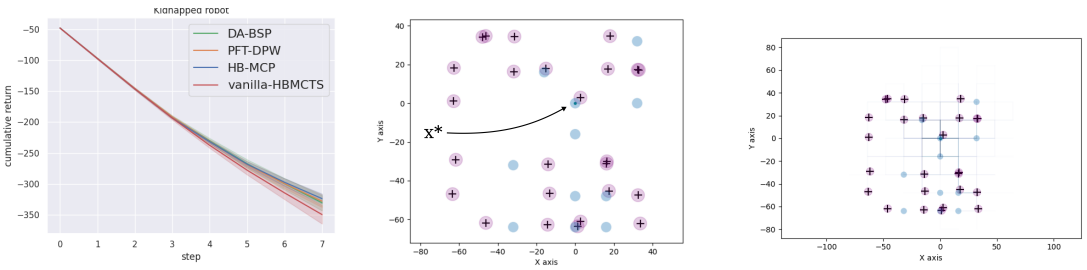


(a) cumulative return (b) initial belief (c) trajectories

Figure A.2: The goal of the agent is to reach the target location while minimizing uncertainty. (a) Mean and standard deviation of the cumulative reward, over 100 trials. (b) Illustration of the initial belief of the agent. x^* denotes the ground truth pose of the agent. l^* denotes a unique landmark. The agent receives as a prior three hypotheses at different locations. (c) Ground-truth trajectories are visualized in transparent color, illustrated on top of the initial belief, such that multiple similar trajectories appear in a moreopaque color.

Goal reaching. As shown in A.2c, most of the trajectories performed by the agent only walk through a simple straight line. Due to the multi-modal hypotheses, the agent first prioritizes the unique landmark (figure A.2b), which practically disambiguates wrong hypotheses due to their large distance from the unique landmark. Then, the agent chooses to reach the goal region to maximize the cumulative reward.

Kidnapped robot. The trajectories shown in figure A.3c do not show a strong preference to any direction. Note that the environment is highly aliased, and there is no unique landmark where the agent may reach to easily disprove wrong hypotheses. Similar results were obtained through all solvers (figure A.3a). Although all landmarks



(a) cumulative return (b) initial belief (c) trajectories

Figure A.3: The goal of the agent is to minimize the uncertainty of its pose. (a) Mean and standard deviation of the cumulative reward, over 100 trials. (b) Illustration of the initial belief of the agent, blue circles illustrate conditional beliefs, crosses denote landmarks. (c) Ground-truth trajectories are visualized in transparent color, illustrated on top of the initial belief, such that multiple similar trajectories appear in a moreopaque color.

look alike, disambiguation may occur by utilizing the pattern of the scattered landmarks. However, such disambiguation may require a long planning horizon which was out of reach for our non-optimized planner.

Hyperparameter	**Description**	**Default Value**
c	UCB exploration constant	40
N_x	Number of state particles per belief node	200
T_m	Time limit per planning step (in seconds)	20^2 / 40^3
$\mathcal{T}$	Lookahead horizon	8
k_o	Observation double progressive widening multiplicative	2.0
α_o	Observation double progressive widening exponent	0.014

Table A.2: Hyperparameters for HB-MCP (ours), vanilla-HB-MCTS and PFT-DPW algorithm. [1] indicates the planning time for Goal reaching and Kidnapped robot scenarios. [2] indicates the planning time for Aliased matrix scenario.

	Aliased matrix	**Goal reaching**	**Kidnapped robot**
HB-MCP (ours)	-585.2	-716.8	-323.7
vanilla-HB-MCTS	-909.6	-939.4	-349.5
PFT-DPW	-961.8	-1009.8	-327.8
DA-BSP	-979.5	-931.5	-330.4

Table A.3: Comparison of algorithm performances on different scenarios. Results are based on a simulation study with 100 trials per scenario and algorithm.

A.4 Data Association Aware POMDP Planning with Hypothesis Pruning Performance Guarantees

A.4.1 Theoretical analysis

Theorem 1

Theorem A.1. *Let time-step $t = 0$ denote the root of the planning tree. Then, the expected reward for the pruned POMDP, $\overline{M}$, is bounded with respect to the full POMDP, M, through the factor of the pruned weight values, and the maximum immediate reward,*

$$\left|\mathbb{E}[r(b_t, a_t)] - \overline{\mathbb{E}}[r(\bar{b}_t, a_t)]\right| \leq \mathcal{R}_{max}\left[\delta_0^{\beta} + \sum_{\tau=1}^{t-1} \overline{\mathbb{E}}_{z_{1:\tau}}\left[\delta_\tau^{\beta}\right]\right], \tag{A.62}$$

where $\delta_\tau^{\beta} \triangleq \sum_{\beta_\tau \in D_\tau \setminus \overline{D}_\tau} \overline{\mathbb{P}}(\beta_\tau \mid H_\tau)$, i.e. the sum of pruned hypotheses weights at time-step τ.

Proof.. Denote D_t as the total number of new associations at time t, and $\overline{D}t$ as a subset thereof. By definition of the expected future reward,

$$\left|\mathbb{E}[r(b_t)] - \overline{\mathbb{E}}[r(\bar{b}_t)]\right| \tag{A.63}$$

$$= \Big| \int_{z_{1:t}} \int_{x_{0:t}} r_x(x_t) \cdot [b_0 \prod_{\tau=1}^{t} \sum_{\beta_\tau}^{|D_\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau)\mathbb{P}(\beta_\tau \mid x_\tau)\mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.64}$$

$$- \bar{b}_0 \prod_{\tau=1}^{t} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau)\mathbb{P}(\beta_\tau \mid x_\tau)\mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right))] \Big| \tag{A.65}$$

$$\tag{A.66}$$

by marginalizing out the variables β_t, z_t,

$$\Big| \int_{z_{1:t}} \int_{x_{0:t}} r_x(x_t) \cdot \Big[b_0 \mathbb{P}\left(x_t \mid x_{t-1}, \pi_{t-1}\right)) \tag{A.67}$$

$$\cdot \prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|D_\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.68}$$

$$- \bar{b}_0 \mathbb{P}\left(x_t \mid x_{t-1}, \pi_{t-1}\right)) \tag{A.69}$$

$$\cdot \prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \Big] \Big| \tag{A.70}$$

$$\leq \int_{z_{1:t}} \int_{x_{0:t}} \Big| r_x(x_t) \cdot \Big[b_0 \mathbb{P}\left(x_t \mid x_{t-1}, \pi_{t-1}\right)) \cdot \tag{A.71}$$

$$\prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|D_\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.72}$$

$$- \bar{b}_0 \mathbb{P}\left(x_t \mid x_{t-1}, \pi_{t-1}\right)) \cdot \tag{A.73}$$

$$\prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \Big] \Big| \tag{A.74}$$

from Holder's inequality,

$$\leq \mathcal{R}_{max} \int_{z_{1:t}} \int_{x_{0:t}} \Big| \mathbb{P}\left(x_t \mid x_{t-1}, \pi_{t-1}\right)) \tag{A.75}$$

$$\Big[b_0 \cdot \prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|D_\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.76}$$

$$- \bar{b}_0 \cdot \prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \Big] \Big| \tag{A.77}$$

since the transition model is positive, we take out of the absolute operator and marginalize it out,

$$\leq \mathcal{R}_{max} \int_{z_{1:t}} \int_{x_{0:t}} \Big| b_0 \cdot \prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|D_\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.78}$$

$$- \bar{b}_0 \cdot \prod_{\tau=1}^{t-1} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau) \mathbb{P}(\beta_\tau \mid x_\tau) \mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \Big| \tag{A.79}$$

to avoid clutter convenience, we denote

$$\tilde{b}_t \triangleq b_0 \prod_{\tau=1}^{t} \sum_{\beta_\tau}^{|D_\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau)\mathbb{P}(\beta_\tau \mid x_\tau)\mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.80}$$

$$\tilde{\bar{b}}_t \triangleq \bar{b}_0 \prod_{\tau=1}^{t} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau)\mathbb{P}(\beta_\tau \mid x_\tau)\mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)). \tag{A.81}$$

Then we can rewrite it as,

$$\begin{aligned} &= \mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \\ &\Big|\tilde{b}_{t-2} \sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1})\mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2}) \\ &- \tilde{\bar{b}}_{t-2} \sum_{\beta_{t-1}}^{|\overline{D}t-1|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1})\mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})\Big| \end{aligned} \tag{A.82}$$

note how this expression can also be written as, $\mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \left|\tilde{b}_t - \tilde{\bar{b}}_t\right|$. This will be useful for a recursive structure to be discussed later.
We now add and subtract,

$$\begin{aligned} &= \mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \\ &\Big|\tilde{b}_{t-2} \sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1})\mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2}) \\ &- \tilde{\bar{b}}_{t-2} \sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1})\mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2}) \\ &+ \tilde{\bar{b}}_{t-2} \sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1})\mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2}) \\ &- \tilde{\bar{b}}_{t-2} \sum_{\beta_{t-1}}^{|\overline{D}t-1|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1})\mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})\Big| \end{aligned} \tag{A.83}$$

grouping terms and applying triangle inequality,

$$= \mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \tag{A.84}$$

$$\Big| \Big[\tilde{b}_{t-2} - \tilde{\tilde{b}}_{t-2}\Big] \sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \Big| \tag{A.85}$$

$$+ \mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}}$$

$$\Big| \tilde{\tilde{b}}_{t-2} \cdot \Big[\sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})$$

$$- \sum_{\beta_{t-1}}^{|\overline{D}t-1|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2}) \Big] \Big|$$

The first summand describes the loss due to pruning in past time steps. The second summand describes the loss due to pruning at the latest time step. Focusing on the second summand, recall that $\overline{D}t \subseteq D_t$, thus,

$$\mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \tag{A.86}$$

$$\Big| \tilde{\tilde{b}}_{t-2} \cdot \Big[\sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \tag{A.87}$$

$$- \sum_{\beta_{t-1}}^{|\overline{D}t-1|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \Big] \Big| \tag{A.88}$$

$$= \mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \tilde{\tilde{b}}_{t-2} \cdot \tag{A.89}$$

$$\Big| \sum_{\beta_{t-1} \in \overline{D}t-1} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \tag{A.90}$$

$$+ \sum_{\beta_{t-1} \in D_t \setminus \overline{D}t-1} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \tag{A.91}$$

$$- \sum_{\beta_{t-1}}^{|\overline{D}t-1|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \Big| \tag{A.92}$$

$$= \mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \tilde{\tilde{b}}_{t-2} \cdot \tag{A.93}$$

$$\Big| \sum_{\beta_{t-1} \in D_{t-1} \setminus \overline{D}t-1} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \Big| \tag{A.94}$$

since all terms within the absolute operator are positive, we can now drop it entirely,

we then marginalize the observation at time $t-1$,

$$= \mathcal{R}_{max} \int_{z_{1:t-2}} \int_{x_{0:t-1}} \bar{b}_0 \cdot \prod_{\tau=1}^{t-2} \sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau)\mathbb{P}(\beta_\tau \mid x_\tau)\mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right)) \tag{A.95}$$

$$\sum_{\beta_{t-1} \in \neg\overline{D}t-1} \mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}\left(x_{t-1} \mid x_{t-2}, \pi_{t-2}\right)) \tag{A.96}$$

by introducing back the normalizer of the pruned belief, $\mathbb{P}(z_\tau \mid H_\tau^-)$, we get,

$$= \mathcal{R}_{max} \int_{z_{1:t-2}} \prod_{k=1}^{t-2} \overline{\mathbb{P}}\left(z_k \mid H_k^-\right) \int_{x_{0:t-1}} \tag{A.97}$$

$$\bar{b}_0 \cdot \prod_{\tau=1}^{t-2} \left[\frac{\sum_{\beta_\tau}^{|\overline{D}\tau|} \mathbb{P}(z_\tau \mid x_\tau, \beta_\tau)\mathbb{P}(\beta_\tau \mid x_\tau)\mathbb{P}\left(x_\tau \mid x_{\tau-1}, \pi_{\tau-1}\right))}{\overline{\mathbb{P}}\left(z_\tau \mid H_\tau^-\right)} \right] \tag{A.98}$$

$$\sum_{\beta_{t-1} \in \neg\overline{D}t-1} \mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}\left(x_{t-1} \mid x_{t-2}, \pi_{t-2}\right)) \tag{A.99}$$

$$= \mathcal{R}_{max} \int_{z_{1:t-2}} \prod_{k=1}^{t-2} \overline{\mathbb{P}}\left(z_k \mid H_k^-\right) \int_{x_{t-2:t-1}} \bar{b}_{t-2} \sum_{\beta_{t-1} \in \neg\overline{D}t-1} \mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}\left(x_{t-1} \mid x_{t-2}, \pi_{t-2}\right)) \tag{A.100}$$

or, equivalently,

$$= \mathcal{R}_{max} \overline{\mathbb{E}}_{z_{1:t-2}} \left[\int_{x_{t-2:t-1}} \bar{b}_{t-2} \sum_{\beta_{t-1} \in \neg\overline{D}t-1} \mathbb{P}(\beta_{t-1} \mid x_{t-1})\mathbb{P}\left(x_{t-1} \mid x_{t-2}, \pi_{t-2}\right)) \right]. \tag{A.101}$$

Crucially, note how the following term depends only on the survived hypotheses (no access to the pruned hypotheses is required). Finally, by rearranging and marginalizing state variables, we get,

$$= \mathcal{R}_{max} \overline{\mathbb{E}}_{z_{1:t-2}} \left[\int_{x_{t-1}} \sum_{\beta_{t-1} \in \neg\overline{D}t-1} \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \overline{\mathbb{P}}\left(x_{t-1} \mid H_{t-1}^-\right) \right] \tag{A.102}$$

$$= \mathcal{R}_{max} \overline{\mathbb{E}}_{z_{1:t-2}} \left[\sum_{\beta_{t-1} \in \neg\overline{D}t-1} \overline{\mathbb{P}}\left(\beta_{t-1} \mid H_{t-1}^-\right) \right] \tag{A.103}$$

$$= \mathcal{R}_{max} \overline{\mathbb{E}}_{z_{1:t-2}} \left[\sum_{\beta_{t-1} \in \neg\overline{D}t-1} \int_{z_{t-1}} \overline{\mathbb{P}}(\beta_{t-1} \mid H_{t-1}) \overline{\mathbb{P}}\left(z_{t-1} \mid H_{t-1}^-\right) \right] \tag{A.104}$$

$$= \mathcal{R}_{max} \overline{\mathbb{E}}_{z_{1:t-1}} \left[\sum_{\beta_{t-1} \in \neg\overline{D}t-1} \overline{\mathbb{P}}(\beta_{t-1} \mid H_{t-1}) \right] \tag{A.105}$$

Going back to the first summand from equation (A.83) and applying triangle inequality,

we have that,

$$\mathcal{R}_{max} \int_{z_{1:t-1}} \int_{x_{0:t-1}} \tag{A.106}$$

$$\Big| \Big[\tilde{b}_{t-2} - \tilde{\bar{b}}_{t-2}\Big] \sum_{\beta_{t-1}}^{|D_{t-1}|} \mathbb{P}(z_{t-1} \mid x_{t-1}, \beta_{t-1}) \mathbb{P}(\beta_{t-1} \mid x_{t-1}) \mathbb{P}(x_{t-1} \mid x_{t-2}, \pi_{t-2})) \Big| \tag{A.107}$$

$$\leq \mathcal{R}_{max} \int_{z_{1:t-2}} \int_{x_{0:t-2}} \Big| \tilde{b}_{t-2} - \tilde{\bar{b}}_{t-2} \Big| \tag{A.108}$$

recall the recursive structure from equation (A.82), thus,

$$\Big| \mathbb{E}[r(b_t)] - \overline{\mathbb{E}}[r(\bar{b}_t)] \Big| \leq \tag{A.109}$$

$$\mathcal{R}_{max} \overline{\mathbb{E}}_{z_{1:t-1}} \left[\sum_{\beta_{t-1} \in \neg \overline{D} t-1} \overline{\mathbb{P}}(\beta_{t-1} \mid H_{t-1}) \right] + \mathcal{R}_{max} \int_{z_{1:t-2}} \int_{x_{0:t-2}} \Big| \tilde{b}_{t-2} - \tilde{\bar{b}}_{t-2} \Big| \tag{A.110}$$

$$\leq \mathcal{R}_{max} \Big(\overline{\mathbb{E}}_{z_{1:t-1}} \left[\sum_{\beta_{t-1} \in \neg \overline{D} t-1} \overline{\mathbb{P}}(\beta_{t-1} \mid H_{t-1}) \right] \tag{A.111}$$

$$+ \overline{\mathbb{E}}_{z_{1:t-2}} \left[\sum_{\beta_{t-2} \in \neg \overline{D} t-2} \overline{\mathbb{P}}(\beta_{t-2} \mid H_{t-2}) \right] + \int_{z_{1:t-3}} \int_{x_{0:t-3}} \Big| \tilde{b}_{t-3} - \tilde{\bar{b}}_{t-3} \Big| \Big) \leq ... \tag{A.112}$$

$$\leq \mathcal{R}_{max} \left(\sum_{\tau=1}^{t-1} \overline{\mathbb{E}}_{z_{1:\tau}} \left[\sum_{\beta_\tau \in \neg \overline{D} \tau} \overline{\mathbb{P}}(\beta_\tau \mid H_\tau) \right] + \int_{x_0} \Big| b_0 - \bar{b}_0 \Big| dx_0 \right) \tag{A.113}$$

$$\equiv \mathcal{R}_{max} \left(\sum_{\tau=1}^{t-1} \overline{\mathbb{E}}_{z_{1:\tau}} \Big[\delta^\beta(H_\tau) \Big] + \delta_0^\beta \right) \tag{A.114}$$

which concludes our derivation. ■

Corollary 1.1

Corollary A.2. *Without loss of generality, assume that the time step at the root node of the planning tree is $t = 0$. Then, for any policy π, the following holds,*

$$\Big| V^\pi(b_0) - \bar{V}^\pi(\bar{b}_0) \Big| \leq \mathcal{R}_{max} \left[\mathcal{T} \cdot \delta_0^\beta + \sum_{k=1}^{\mathcal{T}} \sum_{\tau=1}^{k} \overline{\mathbb{E}}_{z_{1:\tau}} \Big[\delta_\tau^\beta \Big] \right]. \tag{A.115}$$

Proof. The proof is a direct consequence of the linearity of expectation. ■

Self Normalized Importance Sampling Estimator

In this subsubsection we will derive the SN estimator. The theoretical expected reward at time step t may be written as,

$$\mathbb{E}_{z_{1:t}}[r(b_t)] = \int_{z_{1:t}} \prod_{\tau=1}^{t} \mathbb{P}\left(z_\tau \mid H_\tau^-\right) \sum_{\beta_{0:t} \in D_{0:t}} \mathbb{P}(\beta_{0:t} \mid H_t) \int_{x_t} \mathbb{P}(x_t \mid \beta_{0:t}, H_t) r_x(x_t) \tag{A.116}$$

where $D_{0:t}$ is the set of all hypotheses at time step t. Applying Bayes rule followed by a chain rule on $\mathbb{P}(\beta_{0:t} \mid H_t)$,

$$\int_{z_{1:t}} \prod_{\tau=1}^{t} \mathbb{P}\left(z_\tau \mid H_\tau^-\right) \sum_{\beta_{0:t} \in D_{0:t}} \frac{\mathbb{P}\left(z_t, \beta_t \mid \beta_{0:t-1}, H_t^-\right)}{\mathbb{P}\left(z_t \mid H_t^-\right)} \mathbb{P}(\beta_{0:t-1} \mid H_{t-1}) \int_{x_t} \mathbb{P}(x_t \mid \beta_{0:t}, H_t) r_x(x_t) \tag{A.117}$$

applying this step repeatedly on $\mathbb{P}(\beta_{0:\tau} \mid H_\tau)\ \forall \tau \in [1, t-1]$ results in,

$$\int_{z_{1:t}} \prod_{\tau=1}^{t} \mathbb{P}\left(z_\tau \mid H_\tau^-\right) \sum_{\beta_{0:t} \in D_{0:t}} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \frac{\mathbb{P}\left(z_\tau, \beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right)}{\mathbb{P}\left(z_\tau \mid H_\tau^-\right)} \int_{x_t} \mathbb{P}(x_t \mid \beta_{0:t}, H_t) r_x(x_t) \tag{A.118}$$

$$= \int_{z_{1:t}} \sum_{\beta_{0:t} \in D_{0:t}} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(z_\tau, \beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \int_{x_t} \mathbb{P}(x_t \mid \beta_{0:t}, H_t) r_x(x_t) \tag{A.119}$$

$$= \sum_{\beta_0 \in D_0} \mathbb{P}(\beta_0) \sum_{\beta_1 \in D_1} \mathbb{P}\left(\beta_1 \mid \beta_0, H_1^-\right) \int_{z_1} \mathbb{P}\left(z_1 \mid \beta_{0:1}, H_1^-\right) \cdots \tag{A.120}$$

$$\cdots \sum_{\beta_t \in D_t} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \int_{z_t} \mathbb{P}\left(z_t \mid \beta_{0:t}, H_t^-\right) \int_{x_t} \mathbb{P}(x_t \mid \beta_{0:t}, H_t) r_x(x_t)$$

where the second equality is due to chain rule on $\mathbb{P}\left(z_\tau, \beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right)$ and rearranging terms.

According to equation (A.120) we define a self-normalized importance sampling estimator for the expected reward, at time step t, where both the observations and states are sampled,

$$\hat{\mathbb{E}}_{z_{1:t}}[r(\hat{b}_t)] \triangleq \sum_{\beta_0 \in D_0} \sum_{\beta_1 \in D_1} \sum_{c_1} \cdots \sum_{\beta_t \in D_t} \sum_{z_\tau^c} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \hat{r}\left(b_t^\beta\right) \tag{A.121}$$

$$= \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \hat{r}\left(b_t^\beta\right) \tag{A.122}$$

where $\omega\left(z_\tau\right) = \frac{\mathbb{P}\left(z_\tau \mid \beta_{0:\tau}, H_\tau^-\right)}{Q\left(z_\tau \mid H_\tau^-\right)}$ and $Q(.)$ is the proposal distribution according to which the sampling-based estimator generates observations. Similarly, we define the *pruned*

estimator, where the only difference is the summation over a pruned subset of the hypotheses, denoted $\overline{D}$,

$$\hat{\overline{\mathbb{E}}}_{z_{1:t}}\left[r\left(\hat{\overline{b}}_t\right)\right] = \sum_{z^c_{1:t}} \sum_{\beta_{0:t}\in\overline{D}_{0:t}} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right) \frac{\omega\left(z^c_\tau\right)}{\sum_{z^k_\tau}\omega\left(z^k_\tau\right)} \hat{r}\left(b^\beta_t\right). \quad \text{(A.123)}$$

Theorem 2

Theorem A.3. *Let π be the policy, then the expected reward for the estimated pruned POMDP, $\hat{\overline{M}}$, is bounded with respect to the estimated full POMDP, $\hat{M}$, as follows,*

$$\left|\hat{\mathbb{E}}^\pi_{z_{1:t}}[r(\hat{b}_t)] - \hat{\overline{\mathbb{E}}}^\pi_{z_{1:t}}\left[r\left(\hat{\overline{b}}_t\right)\right]\right| \leq \mathcal{R}_{max}\left[\hat{\delta}^\beta_0 + \sum_{\tau=1}^{t}\hat{\delta}^\beta_\tau\right]. \quad \text{(A.124)}$$

where, $\hat{\delta}^\beta_\tau = \hat{\mathbb{E}}_{z^c_{1:t}}\overline{\mathbb{E}}_{\beta_{0:t-1}}\sum_{\beta_t\in D_t\setminus\overline{D}_t}\mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H^-_t\right)$ for all $\tau\in[1,t]$ represents the expected sum of conditional hypotheses' weights which are myopically pruned and $\hat{\delta}^\beta_0 = \sum_{\beta_0\in D_0\setminus\overline{D}_0}\mathbb{P}\left(\beta_0 \mid H^-_t\right)$.

Proof.. Hereon forward, we assume that conditioned the same hypothesis, $\beta_{0:\tau}$, the same observations and states are sampled. This is required in order to obtain a deterministic bound and can be achieved in practice by fixing some seed number. Additionally, we also define a pruned conditionals,

$$\overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right) \triangleq \begin{cases} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right) & ,\ \beta_{0:\tau}\in\overline{D}_{0:\tau} \\ 0 & ,\ otherwise \end{cases}. \quad \text{(A.125)}$$

Then,

$$\left|\hat{\mathbb{E}}_{z_{1:t}}[r(\hat{b}_t)] - \hat{\overline{\mathbb{E}}}_{z_{1:t}}\left[r\left(\hat{\overline{b}}_t\right)\right]\right| \quad \text{(A.126)}$$

$$= \Big|\sum_{\beta_0\in D_0}\prod_{k=1}^{t}\sum_{\beta_k\in D_k}\sum_{z^c_\tau}\mathbb{P}(\beta_0)\prod_{\tau=1}^{t}\mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right)\frac{\omega(z^c_\tau)}{\sum_{z^k_\tau}\omega(z^k_\tau)}\hat{r}\left(b^\beta_t\right) \quad \text{(A.127)}$$

$$- \sum_{\beta_0\in\overline{D}_0}\prod_{k=1}^{t}\sum_{\beta_k\in\overline{D}_k}\sum_{z^c_\tau}\mathbb{P}(\beta_0)\prod_{\tau=1}^{t}\mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right)\frac{\omega(z^c_\tau)}{\sum_{z^k_\tau}\omega(z^k_\tau)}\hat{r}\left(b^\beta_t\right)\Big|$$

$$= \Big|\sum_{\beta_0\in D_0}\prod_{k=1}^{t}\sum_{\beta_k\in D_k}\sum_{z^c_\tau}\mathbb{P}(\beta_0)\prod_{\tau=1}^{t}\mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right)\frac{\omega(z^c_\tau)}{\sum_{z^k_\tau}\omega(z^k_\tau)}\hat{r}\left(b^\beta_t\right) \quad \text{(A.128)}$$

$$- \sum_{\beta_0\in D_0}\prod_{k=1}^{t}\sum_{\beta_k\in D_k}\sum_{z^c_\tau}\overline{\mathbb{P}}(\beta_0)\prod_{\tau=1}^{t}\overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H^-_\tau\right)\frac{\omega(z^c_\tau)}{\sum_{z^k_\tau}\omega(z^k_\tau)}\hat{r}\left(b^\beta_t\right)\Big|$$

add and subtract,

$$
\begin{aligned}
&| \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \hat{r}\left(b_t^\beta\right) \qquad \text{(A.129)} \\
&- \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \frac{\omega_t^c}{\sum_{c_t'} \omega_t^{c'}} \hat{r}\left(b_t^\beta\right) \\
&+ \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \frac{\omega_t^c}{\sum_{c_t'} \omega_t^{c'}} \hat{r}\left(b_t^\beta\right) \\
&- \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \hat{r}\left(b_t^\beta\right) |
\end{aligned}
$$

applying triangle inequality then focusing on the second pair of terms,

$$
| \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_t^k} \omega(z_t^k)} \hat{r}\left(b_t^\beta\right) \qquad \text{(A.130)}
$$

$$
\begin{aligned}
&- \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \hat{r}\left(b_t^\beta\right) | \\
&= | \sum_{\beta_0 \in D_0} \prod_{k=1}^{t-1} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \cdot \qquad \text{(A.131)} \\
&\sum_{\beta_t \in D_t} \left[\mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) - \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right)\right] \sum_{z_\tau^c} \frac{\omega(z_\tau^c)}{\sum_{z_t^k} \omega(z_t^k)} \hat{r}\left(b_t^\beta\right) |
\end{aligned}
$$

applying again triangle inequality followed by Holder inequality,

$$
\begin{aligned}
&\leq \mathcal{R}_{max} \sum_{\beta_0 \in D_0} \prod_{k=1}^{t-1} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \cdot \qquad \text{(A.132)} \\
&\left| \sum_{\beta_t \in D_t} \left[\mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) - \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right)\right] \right| \\
&= \mathcal{R}_{max} \sum_{\beta_0 \in D_0} \prod_{k=1}^{t-1} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \cdot \qquad \text{(A.133)} \\
&\quad \sum_{\beta_t \in D_t \setminus \overline{D}_t} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \\
&\triangleq \mathcal{R}_{max} \hat{\delta}_t^\beta
\end{aligned}
$$

where $\hat{\delta}_t^\beta$ is the empirical expected weight of all the pruned hypotheses at time step t. Crucially, its value depends only on past pruned hypotheses, which are known to us. Now focusing on the first pair of terms from equation (A.129),

$$| \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \mathbb{P}(\beta_0) \prod_{\tau=1}^{t} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \hat{r}\left(b_t^\beta\right) \tag{A.134}$$

$$- \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \frac{\omega(z_t^c)}{\sum_{z_t^k} \omega(z_t^k)} \hat{r}\left(b_t^\beta\right) |$$

$$= | \sum_{\beta_0 \in D_0} \prod_{k=1}^{t} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} \left[\mathbb{P}(\beta_0) \prod_{\tau=1}^{t-1} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} - \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \right] \cdot \tag{A.135}$$

$$\mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \frac{\omega(z_t^c)}{\sum_{z_t^k} \omega(z_t^k)} \hat{r}\left(b_t^\beta\right) |$$

triangle and Holder inequalities,

$$\leq \sum_{\beta_0 \in D_0} \prod_{k=1}^{t-1} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} |\mathbb{P}(\beta_0) \prod_{\tau=1}^{t-1} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \tag{A.136}$$

$$- \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} | \sum_{\beta_t \in D_t} \sum_{z_\tau^c} \mathbb{P}\left(\beta_t \mid \beta_{0:t-1}, H_t^-\right) \frac{\omega(z_t^c)}{\sum_{z_t^k} \omega(z_t^k)} \mathcal{R}_{max}$$

$$= \mathcal{R}_{max} \sum_{\beta_0 \in D_0} \prod_{k=1}^{t-1} \sum_{\beta_k \in D_k} \sum_{z_\tau^c} |\mathbb{P}(\beta_0) \prod_{\tau=1}^{t-1} \mathbb{P}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} \tag{A.137}$$

$$- \overline{\mathbb{P}}(\beta_0) \prod_{\tau=1}^{t-1} \overline{\mathbb{P}}\left(\beta_\tau \mid \beta_{0:\tau-1}, H_\tau^-\right) \frac{\omega(z_\tau^c)}{\sum_{z_\tau^k} \omega(z_\tau^k)} |$$

then, applying similar steps recursively on the obtained term yields,

$$\left| \hat{\mathbb{E}}_{z_{1:t}}[r(\hat{b}_t)] - \hat{\overline{\mathbb{E}}}_{z_{1:t}}\left[r\left(\hat{\overline{b}}_t\right)\right] \right| \leq \mathcal{R}_{max} \sum_{\tau=0}^{t} \hat{\delta}_\tau^\beta \tag{A.138}$$

which concludes our derivation. ■

Corollary 2.1

Corollary A.4. *The difference between the estimated value function of the full POMDP, $\hat{M}$, and the estimated value function of the pruned POMDP, $\hat{\overline{M}}$, is bounded by,*

$$|\hat{V}^\pi(\hat{b}_0) - \hat{\overline{V}}^\pi(\hat{\overline{b}}_0)| \leq \mathcal{R}_{max} \left[\hat{\delta}_0^\beta + \sum_{k=1}^{\mathcal{T}} \sum_{\tau=1}^{k} \hat{\delta}_\tau^\beta \right]. \tag{A.139}$$

Proof. The proof is a direct consequence of the linearity of expectation. ■

Corollary 2.2

Corollary A.5. *Let π be a policy and let $\mathcal{A}$ be a sampling-based estimator for the value function such that $|V^{\pi}(b_0) - \hat{V}^{\pi}(\hat{b}_0)| \leq \epsilon_{\mathcal{A}}$ with probability at least $1 - \delta_{\mathcal{A}}$. Then, the following corollary holds for the loss in the value function for the pruned hypotheses,*

$$|V^{\pi}(b_0) - \hat{\bar{V}}^{\pi}(\hat{\bar{b}}_0)| \leq \tag{A.140}$$

$$|V^{\pi}(b_0) - \hat{V}^{\pi}(\hat{b}_0)| + |\hat{V}^{\pi}(\hat{b}_0) - \hat{\bar{V}}^{\pi}(\hat{\bar{b}}_0)| \leq \epsilon_{\mathcal{A}} + \hat{\epsilon}^{hs}_{\bar{D}}, \tag{A.141}$$

hold with probability $1 - \delta_{\mathcal{A}}$. We use $\hat{\epsilon}^{hs}_{\bar{D}}$ as a shorthand for the bounds provided in corollary A.4.

Corollary 2.3

Corollary A.6. *Let $\pi^{\star}$ be the optimal policy for the full theoretical POMDP with a respective value function, $V(b_t)$. Let $\bar{\pi}$ be the optimal policy for the pruned POMDP and a value function, $\bar{V}(\bar{b}_t)$. Last, let $\hat{\bar{\pi}}$ be the optimal policy for the pruned, sampled-based POMDP with a value function, $\hat{\bar{V}}(\hat{\bar{b}}_t)$. Then, with probability at least $1 - \delta_{\mathcal{A}}$, the following holds,*

$$\left|V^{\pi^{\star}}(b_t) - \hat{\bar{V}}^{\hat{\bar{\pi}}}(\hat{\bar{b}}_t)\right| \leq 2(\epsilon_{\mathcal{A}} + \hat{\epsilon}^{hs}_{\bar{D}}). \tag{A.142}$$

Proof.. For conciseness, we drop the explicit dependence on the belief at each value function.

$$\left|V^{\pi^*} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \leq \underbrace{\left|V^{\pi^*} - \hat{V}^{\hat{\pi}}\right|}_{(a)} + \underbrace{\left|\hat{V}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right|}_{(b)} \tag{A.143}$$

we split the derivation of term (a) into two cases, case 1a: $V^{\hat{\pi}} \geq \hat{V}^{\pi^*}$, then, $V^{\pi^*} \geq V^{\hat{\pi}} \geq \hat{V}^{\pi^*}$

$$\begin{aligned}(a) = \left|V^{\pi^*} - \hat{V}^{\hat{\pi}}\right| &\leq \left|V^{\pi^*} - V^{\hat{\pi}}\right| + \left|V^{\hat{\pi}} - \hat{V}^{\hat{\pi}}\right| \\ &\leq \left|V^{\pi^*} - \hat{V}^{\pi^*}\right| + \left|V^{\hat{\pi}} - \hat{V}^{\hat{\pi}}\right| \leq 2\epsilon_{\mathcal{A}}\end{aligned} \tag{A.144}$$

Case 2a: $V^{\hat{\pi}} \leq \hat{V}^{\pi^*}$. Then, $V^{\hat{\pi}} \leq \hat{V}^{\pi^*} \leq \hat{V}^{\hat{\pi}}$. By triangle inequality,

$$(a) = \left|V^{\pi^*} - \hat{V}^{\hat{\pi}}\right| \leq \left|V^{\pi^*} - \hat{V}^{\pi^*}\right| + \left|\hat{V}^{\pi^*} - \hat{V}^{\hat{\pi}}\right| \tag{A.145}$$

$$\leq \left|V^{\pi^*} - \hat{V}^{\pi^*}\right| + \left|V^{\hat{\pi}} - \hat{V}^{\hat{\pi}}\right| \leq 2\epsilon_{\mathcal{A}} \tag{A.146}$$

Similarly, we split the handling of term (b) to two different cases, case 1b: if $\hat{V}^{\hat{\bar{\pi}}} \geq \hat{\bar{V}}^{\hat{\pi}}$,

then, $\hat{V}^{\hat{\pi}} \geq \hat{V}^{\hat{\bar{\pi}}} \geq \hat{\bar{V}}^{\hat{\bar{\pi}}}$. By triangle inequality,

$$(b) = \left|\hat{V}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \leq \left|\hat{V}^{\hat{\pi}} - \hat{V}^{\hat{\bar{\pi}}}\right| + \left|\hat{V}^{\hat{\bar{\pi}}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \tag{A.147}$$

$$\leq \left|\hat{V}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\pi}}\right| + \left|\hat{V}^{\hat{\bar{\pi}}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \leq 2\hat{\epsilon}_{\bar{D}}^{hs} \tag{A.148}$$

where the last inequality is due to corollary A.5. Case 2b: if $\hat{V}^{\hat{\bar{\pi}}} \leq \hat{\bar{V}}^{\hat{\pi}}$, then,$\hat{V}^{\hat{\bar{\pi}}} \leq \hat{\bar{V}}^{\hat{\pi}} \leq \hat{\bar{V}}^{\hat{\bar{\pi}}}$. From triangle inequality,

$$(b) = \left|\hat{V}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \leq \left|\hat{V}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\pi}}\right| + \left|\hat{\bar{V}}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \tag{A.149}$$

$$\leq \left|\hat{V}^{\hat{\pi}} - \hat{\bar{V}}^{\hat{\pi}}\right| + \left|\hat{V}^{\hat{\bar{\pi}}} - \hat{\bar{V}}^{\hat{\bar{\pi}}}\right| \leq 2\hat{\epsilon}_{\bar{D}}^{hs} \tag{A.150}$$

which covers all the cases and result in,

$$\left|V^{\pi^\star}(b_t) - \hat{\bar{V}}^{\hat{\bar{\pi}}}(\hat{\bar{b}}_t)\right| \leq 2(\epsilon_{\mathcal{A}} + \hat{\epsilon}_{\bar{D}}^{hs}). \tag{A.151}$$

A.5 Online POMDP Planning with Anytime Deterministic Guarantees

A.6 Mathematical Analysis

We start by restating the definition of the simplified value function,

$$\bar{V}^{\pi}(\bar{b}_t) \triangleq r(\bar{b}_t, \pi_t) + \bar{\mathbb{E}}\left[\bar{V}(b_t)\right] \tag{A.152}$$

$$= \sum_{x_t} \bar{b}(x_t) r_x(x_t, \pi_t) + \sum_{z_t} \bar{\mathbb{P}}(z_{t+1} \mid H_{t+1}^{-}) \bar{V}(\bar{b}(z_{t+1})), \tag{A.153}$$

A.6.1 Theorem 1

Theorem A.7. *Let b_t belief state at time t, and $\mathcal{T}$ be the last time step of the POMDP. Let $V^{\pi}(b_t)$ be the theoretical value function by following a policy π, and let $\bar{V}^{\pi}(b_t)$ be the simplified value function, as defined in* (5.7)*, by following the same policy. Then, for any policy π, the difference between the theoretical and simplified value functions is bounded as follows,*

$$\left|V^{\pi}(b_t) - \bar{V}^{\pi}(b_t)\right| \leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \triangleq \epsilon_z^{\pi}(b_t). \tag{A.154}$$

Proof.. For notational convenience, we derive the bounds for the value function by denoting the prior belief as b_0,

$$V_0^{\pi}(b_0) = \mathbb{E}_{z_{1:\mathcal{T}}}\left[\sum_{t=0}^{\mathcal{T}} r(b_t, a_t)\right] \tag{A.155}$$

applying the belief update equation,

$$V_0^{\pi}(b_0) = \sum_{z_{1:\mathcal{T}}} \prod_{\tau=1}^{\mathcal{T}} \mathbb{P}\left(z_\tau \mid H_\tau^{-}\right) \sum_{t=0}^{\mathcal{T}} \left[\sum_{x_t} \frac{\mathbb{P}(z_t \mid x_t) \sum_{x_{t-1}} \mathbb{P}(x_t \mid x_{t-1}, \pi_{t-1}) b_{t-1}}{\mathbb{P}\left(z_t \mid H_t^{-}\right)} r_x(x_t, a_t)\right] \tag{A.156}$$

$$= \sum_{z_{1:\mathcal{T}}} \prod_{\tau=1}^{\mathcal{T}} \mathbb{P}\left(z_\tau \mid H_\tau^{-}\right) \sum_{t=0}^{\mathcal{T}} \left[\sum_{x_{0:t}} \frac{\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0)}{\prod_{\tau=1}^{t} \mathbb{P}\left(z_\tau \mid H_\tau^{-}\right)} r_x(x_t, a_t)\right] \tag{A.157}$$

$$= \sum_{t=0}^{\mathcal{T}} \sum_{z_{1:\mathcal{T}}} \sum_{x_{0:\mathcal{T}}} \prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) r_x(x_t, a_t) \tag{A.158}$$

which applies similarly to the simplified value function,

$$\bar{V}_0^{\pi}(b_0) = \sum_{t=0}^{\mathcal{T}} \sum_{z_{1:\mathcal{T}}} \sum_{x_{0:\mathcal{T}}} \prod_{k=1}^{t} \bar{\mathbb{P}}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) r_x(x_t, a_t). \tag{A.159}$$

We begin the derivation by focusing on a single time step, t, and later generalize to the complete value function.

$$|\mathbb{E}_{z_{1:t}}[r(b_t)] - \bar{\mathbb{E}}_{z_{1:t}}[r(\bar{b}_t)]| \tag{A.160}$$

$$= |\sum_{z_{1:t}} \sum_{x_{0:t}} [\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) r_x(x_t) - \prod_{k'=1}^{t} \bar{\mathbb{P}}(z_{k'} \mid x_{k'})\mathbb{P}(x_{k'} \mid x_{k'-1}, \pi_{k'-1}) b(x_0) r_x(x_t)]| \tag{A.161}$$

$$\leq \sum_{z_{1:t}} \sum_{x_{0:t}} \left| r_x(x_t) \left[\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) - \prod_{k'=1}^{t} b(x_0)\ \bar{\mathbb{P}}(z_{k'} \mid x_{k'})\mathbb{P}(x_{k'} \mid x_{k'-1}, \pi_{k'-1}) \right] \right| \tag{A.162}$$

$$= \sum_{z_{1:t}} \sum_{x_{0:t}} |r_x(x_t)| \left[\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) - \prod_{k'=1}^{t} b(x_0)\ \bar{\mathbb{P}}(z_{k'} \mid x_{k'})\ \mathbb{P}(x_{k'} \mid x_{k'-1}, \pi_{k'-1}) \right] \tag{A.163}$$

where the second transition is due to triangle inequality, the third transition is equality by the construction, i.e. using the simplified observation models imply that the difference is nonnegative. We add and subtract, followed by rearranging terms,

$$= \sum_{z_{1:t}} \sum_{x_{0:t}} |r_x(x_t)| \tag{A.164}$$

$$\begin{aligned} &[\prod_{k=1}^{t} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) - \prod_{k=1}^{t-1} b(x_0) \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \\ &+ \prod_{k=1}^{t-1} b(x_0) \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) - \prod_{k'=1}^{t} b(x_0) \bar{\mathbb{P}}(z_{k'}, x_{k'} \mid x_{k'-1}, \pi_{k'-1})] \end{aligned}$$

$$= \sum_{z_{1:t}} \sum_{x_{0:t}} |r_x(x_t)| \Big\{ \tag{A.165}$$

$$\begin{aligned} &\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \left[\prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) - \prod_{k=1}^{t-1} b(x_0) \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] \\ &+ \prod_{k=1}^{t-1} b(x_0) \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) [\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) - \ \bar{\mathbb{P}}(z_t, x_t \mid x_{t-1}, \pi_{t-1})] \Big\} \end{aligned}$$

applying Holder's inequality,

$$\leq \mathcal{R}_{\max} \sum_{z_{1:t}} \sum_{x_{0:t}} \mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \left[b(x_0) \prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) - b(x_0) \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] \tag{A.166}$$

$$+ \mathcal{R}_{\max} \sum_{z_{1:t}} \sum_{x_{0:t}} \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) [\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) - \bar{\mathbb{P}}(z_t, x_t \mid x_{t-1}, \pi_{t-1})]$$

$$= \mathcal{R}_{\max} \sum_{z_{1:t}} \sum_{x_{0:t}} \mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \cdot \tag{A.167}$$

$$\left[b(x_0) \prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) - b(x_0) \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] + \mathcal{R}_{\max} \delta_t$$

$$= \mathcal{R}_{\max} \sum_{z_{1:t-1}} \sum_{x_{0:t-1}} \left[b(x_0) \prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) - b(x_0) \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] \tag{A.168}$$

$$+ \mathcal{R}_{\max} \delta_t,$$

following similar steps recursively,

$$= \ldots = \mathcal{R}_{\max} \sum_{\tau=1}^{t} \delta_\tau. \tag{A.169}$$

Finally, applying similar steps for every time step $t \in [1, \mathcal{T}]$ results in,

$$\left| V^\pi(b_t) - \bar{V}^\pi(b_t) \right| \leq \mathcal{R}_{\max} \sum_{t=1}^{\mathcal{T}} \sum_{\tau=1}^{t} \delta_\tau \tag{A.170}$$

where,

$$\delta_\tau = \sum_{z_{1:\tau}} \sum_{x_{0:\tau}} \prod_{k=1}^{\tau-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) [\mathbb{P}(z_\tau, x_\tau \mid x_{\tau-1}, \pi_{\tau-1}) - \bar{\mathbb{P}}(z_\tau, x_\tau \mid x_{\tau-1}, \pi_{\tau-1})]$$

$$= \sum_{z_{1:\tau-1}} \sum_{x_{0:\tau-1}} \prod_{k=1}^{\tau-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) [1 - \sum_{z_\tau} \sum_{x_\tau} \bar{\mathbb{P}}(z_\tau, x_\tau \mid x_{\tau-1}, \pi_{\tau-1})] \tag{A.171}$$

plugging the term in (A.171) to (A.170) and expanding the terms results in the desired bound,

$$\left| V^\pi(b_t) - \bar{V}^\pi(b_t) \right| \leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \right] \tag{A.172}$$

which concludes our derivation. ■

A.6.2 Lemma 1

Lemma A.6.1. *The optimal value function can be bounded as*

$$V^{\pi *}(b_t) \leq \mathrm{UDB}^{\pi}(b_t), \tag{A.173}$$

where the policy π is determined according to Bellman optimality over the UDB, i.e.

$$\mathrm{UDB}^{\pi}(b_t) \triangleq \max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi}(b_t, a_t) + \epsilon_z^{\pi}(b_t, a_t)] \tag{A.174}$$

$$= \max_{a_t \in \mathcal{A}} [r(b_t, a_t) + \bar{\mathbb{E}}_{z_{t+1}|b_t,a_t}[\bar{V}^{\pi}(b_{t+1})] + \epsilon_z^{\pi}(b_t, a_t)]. \tag{A.175}$$

Proof.. In the following, we prove by induction that applying the Bellman optimality operator on upper bounds to the value function in finite-horizon POMDPs will result in an upper bound on the optimal value function. The notations are the same as the ones presented in chapter 5. We restate some of the definitions from the chapter for convenience.

The policy $\pi_t(b_t)$ determined by applying Bellman optimality at belief b_t, i.e.,

$$\pi_t(b_t) = \arg\max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi}(b_t, a_t) + \epsilon_z^{\pi}(b_t, a_t)]. \tag{A.176}$$

As it will be needed in the following proof, we also define the value of a belief which includes in its history at least one observation out of the simplified set, e.g. $H_t = \{a_0, z_1, \ldots, z_k \notin \bar{\mathcal{Z}}, \ldots, z_t\}$ as being equal to zero. Explicitly,

$$\bar{V}_t^{\pi}(\mathbb{P}(x_t \mid a_0, z_1, \ldots, z_k \notin \bar{\mathcal{Z}}, \ldots, z_t)) \equiv 0 \quad \forall k \in [1, t]. \tag{A.177}$$

We also use the following simple bound,

$$V_{t,\max} \triangleq \mathcal{R}_{\max} \cdot (\mathcal{T} - t - 1) \tag{A.178}$$

Base case ($t = \mathcal{T}$) - At the final time step $\mathcal{T}$, for each belief we set the value function to be equal to the reward value at that belief state, b_T and taking the action that maximizes the immediate reward,

$$\mathrm{UDB}^{\pi}(b_{\mathcal{T}}) = \max_{a_{\mathcal{T}}} \{r(b_{\mathcal{T}}, a_{\mathcal{T}}) + \epsilon_z(b_{\mathcal{T}}, a_{\mathcal{T}})\} \equiv \arg\max_{a_{\mathcal{T}}} \{r(b_{\mathcal{T}}, a_{\mathcal{T}})\} \tag{A.179}$$

which provides an upper bound for the optimal value function for the final time step, $V_{\mathcal{T}}^{\star}(b_{\mathcal{T}}) \leq \mathrm{UDB}^{\pi}(b_{\mathcal{T}})$.

Induction hypothesis - Assume that for a given time step, t, for all belief states the following holds,

$$V_t^{\star}(b_t) \leq \mathrm{UDB}^{\pi}(b_t). \tag{A.180}$$

Induction step - We will show that the hypothesis holds for time step $t-1$. By the induction hypothesis,

$$V_t^\star(b_t) \leq \mathrm{UDB}^\pi(b_t) \quad \forall b_t, \tag{A.181}$$

thus,

$$Q^\star(b_{t-1}, a_{t-1}) = r(b_{t-1}, a_{t-1}) + \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) V_t^\star(b(z_t)) \tag{A.182}$$

$$\leq r(b_{t-1}, a_{t-1}) + \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \mathrm{UDB}^\pi(b(z_t)) \tag{A.183}$$

$$= r(b_{t-1}, a_{t-1}) + \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \left[\bar{V}_t^\pi(b_t) + \epsilon_z^\pi(b_t)\right]. \tag{A.184}$$

For the following transition, we make use of lemma A.6.1,

$$= r(b_{t-1}, a_{t-1}) + \bar{\mathbb{E}}_{z_t | b_{t-1}, a_{t-1}} \left[\bar{V}_t^\pi(b_t)\right] + \epsilon_z^\pi(b_{t-1}, a_{t-1}) \tag{A.185}$$

$$\equiv \mathrm{UDB}^\pi(b_{t-1}, a_{t-1}). \tag{A.186}$$

Therefore, under the induction hypothesis, $Q_{t-1}^\star(b_{t-1}, a_{t-1}) \leq \mathrm{UDB}^\pi(b_{t-1}, a_{t-1})$. Taking the maximum over all actions a_t,

$$\mathrm{UDB}^\pi(b_{t-1}) = \max_{a_{t-1} \in \mathcal{A}} \{\mathrm{UDB}^\pi(b_{t-1}, a_{t-1})\} \tag{A.187}$$

$$\geq \max_{a_{t-1} \in \mathcal{A}} \{Q_{t-1}^\star(b_{t-1}, a_{t-1})\} = V_{t-1}^\star(b_{t-1}),$$

which completes the induction step and the required proof. ■

Lemma A.6.2. *Let b_t denote a belief state and π_t a policy at time t. Let $\bar{\mathbb{P}}(z_t \mid x_t)$ be the simplified observation model which represents the likelihood of observing z_t given x_t. Then, the following terms are equivalent,*

$$\mathbb{E}_{z_t}\left[\bar{V}_t^\pi(b_t) + \epsilon_z^\pi(b_t)\right] = \bar{\mathbb{E}}_{z_t}\left[\bar{V}_t^\pi(b_t)\right] + \epsilon_z^\pi(b_{t-1}, a_{t-1}) \tag{A.188}$$

Proof..

$$\mathbb{E}_{z_t}\left[\bar{V}_t^\pi(b_t) + \epsilon_z^\pi(b_t)\right] = \tag{A.189}$$

$$\mathbb{E}_{z_t}\left[\bar{V}_t^\pi(b_t)\right] + \mathbb{E}_{z_t}\left[\mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b_t \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right]\right] \tag{A.190}$$

focusing on the second summand,

$$\sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b_t \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \tag{A.191}$$

$$= \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_t} \mathbb{P}\left(z_t \mid H_t^-\right) \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \tag{A.192}$$

by marginalizing over x_{t-1},

$$= \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} [1 - \sum_{z_t} \mathbb{P}\left(z_t \mid H_t^-\right) \sum_{z_{t+1:\tau}} \sum_{x_{t-1:\tau}} \frac{\bar{\mathbb{P}}(z_t \mid x_t) \mathbb{P}(x_t \mid x_{t-1}, \pi_{t-1}) b(x_{t-1})}{\mathbb{P}\left(z_t \mid H_t^-\right)} \cdot \tag{A.193}$$

$$\prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})]$$

canceling out the denominator,

$$= \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} [1 - \sum_{z_{t:\tau}} \sum_{x_{t-1:\tau}} \bar{\mathbb{P}}(z_t \mid x_t) \mathbb{P}(x_t \mid x_{t-1}, a_{t-1}) b(x_{t-1}) \cdot \tag{A.194}$$

$$\prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})] \equiv \epsilon_z^{\pi}(b_{t-1}, a_{t-1})$$

it is left to show that $\mathbb{E}_{z_t|b_{t-1},a_{t-1}}\left[\bar{V}_t^{\pi}(b_t)\right] = \bar{\mathbb{E}}_{z_t|b_{t-1},a_{t-1}}\left[\bar{V}_t^{\pi}(b_t)\right]$. By the definition of a value function of a belief not included in the simplified set, we have that,

$$\mathbb{E}_{z_t|b_{t-1},a_{t-1}}\left[\bar{V}_t^{\pi}(b_t)\right] = \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \bar{V}_t^{\pi}(b_t) \tag{A.195}$$

$$= \sum_{z_t \in \bar{\mathcal{Z}}} \mathbb{P}\left(z_t \mid H_t^-\right) \bar{V}_t^{\pi}(b_t) + \sum_{z_t \in \mathcal{Z} \setminus \bar{\mathcal{Z}}} \mathbb{P}\left(z_t \mid H_t^-\right) \bar{V}_t^{\pi}(b_t) \tag{A.196}$$

$$= \sum_{z_t \in \bar{\mathcal{Z}}} \bar{\mathbb{P}}\left(z_t \mid H_t^-\right) \cdot \bar{V}_t^{\pi}(b_t) + \sum_{z_t \in \mathcal{Z} \setminus \bar{\mathcal{Z}}} \mathbb{P}\left(z_t \mid H_t^-\right) \cdot 0 \tag{A.197}$$

$$= \bar{\mathbb{E}}_{z_t|b_{t-1},a_{t-1}}\left[\bar{V}_t^{\pi}(b_t)\right], \tag{A.198}$$

which concludes the derivation. ■

A.6.3 Corollary 1.1

We restate the definition of UDB exploration criteria,

$$a_t = \arg\max_{a_t \in \mathcal{A}} [\text{UDB}^\pi(b_t, a_t)] = \arg\max_{a_t \in \mathcal{A}} [\bar{Q}^\pi(b_t, a_t) + \epsilon_z^\pi(b_t, a_t)]. \tag{A.199}$$

Corollary A.8. *Using Lemma A.6.1 and the exploration criteria defined in* (5.17) *guarantees convergence to the optimal value function.*

Proof.. Let us define a sequence of bounds, $\text{UDB}_n^\pi(b_t)$ and a corresponding difference value between UDB_n and the simplified value function,

$$\text{UDB}_n^\pi(b_t) - \bar{V}_n^\pi(b_t) = \epsilon_{n,z}^\pi(b_t), \tag{A.200}$$

where $n \in [0, |\mathcal{Z}|]$ corresponds to the number of unique observation instances within the simplified observation set, $\bar{\mathcal{Z}}_n$, and $|\mathcal{Z}|$ denotes the cardinality of the complete observation space. Additionally, for the clarity of the proof and notations, assume that by construction the simplified set is chosen such that $\bar{\mathcal{Z}}_n(H_t) \equiv \bar{\mathcal{Z}}_n$ remains identical for all time steps t and history sequences, H_t given n. By the definition of $\epsilon_{n,z}^\pi(b_t)$,

$$\epsilon_{n,z}^\pi(b_t) = \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau} \in \bar{\mathcal{Z}}_n} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \right], \tag{A.201}$$

we have that $\epsilon_{n,z}^\pi(b_t) \to 0$ as $n \to |\mathcal{Z}|$, since

$$\sum_{z_{t+1:\tau} \in \bar{\mathcal{Z}}_n} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \to 1 \tag{A.202}$$

as more unique observation elements are added to the simplified observation space, $\bar{\mathcal{Z}}_n$, eventually recovering the entire support of the discrete observation distribution.

From lemma A.6.1 we have that, for all $n \in [0, |\mathcal{Z}|]$ the following holds,

$$V^{\pi *}(b_t) \leq \text{UDB}_n^\pi(b_t) = \bar{V}_n^\pi(b_t) + \epsilon_{n,z}^\pi(b_t). \tag{A.203}$$

Additionally, from theorem 5.1 we have that,

$$\left| V^\pi(b_t) - \bar{V}_n^\pi(b_t) \right| \leq \epsilon_{n,z}^\pi(b_t), \tag{A.204}$$

for any policy π and subset $\bar{\mathcal{Z}}_n \subseteq \mathcal{Z}$, thus,

$$\bar{V}_n^\pi(b_t) - \epsilon_{n,z}^\pi(b_t) \leq V^\pi(b_t) \leq V^{\pi *}(b_t) \leq \bar{V}_n^\pi(b_t) + \epsilon_{n,z}^\pi(b_t). \tag{A.205}$$

Since $\epsilon_{n,z}^\pi(b_t) \to 0$ as $n \to |\mathcal{Z}|$, and $|\mathcal{Z}|$ is finite, it is guaranteed that $\text{UDB}_n^\pi(b_t) \xrightarrow{n \to |\mathcal{Z}|}$

$V^{\pi*}(b_t)$ which completes our proof. ■

Moreover, depending on the algorithm implementation, the number of iterations can be finite (e.g. by directly choosing actions and observations to minimize the bound). A stopping criteria can also be verified by calculating the difference between the upper and lower bounds. The optimal solution is obtained once the upper bound equals the lower bound.

A.6.4 Theorem 2

Theorem A.9. *Let b_t belief state at time t, and $\mathcal{T}$ be the last time step of the POMDP. Let $V^{\pi}(b_t)$ be the theoretical value function by following a policy π, and let $\bar{V}^{\pi}(b_t)$ be the simplified value function, as defined in* (5.7)*, by following the same policy. Then, for any policy π, the difference between the theoretical and simplified value functions is bounded as follows,*

$$\left|V^{\pi}(b_t) - \bar{V}^{\pi}(b_t)\right| \leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{\mathcal{T}} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \triangleq \epsilon^{\pi}(b_t). \tag{A.206}$$

Recall that we define $\tau_t = \{x_0, a_0, z_1, x_1, a_1, \ldots, a_{\mathcal{T}-1}, x_t, z_t\}$. Then the value function is defined as,

$$V^{\pi}(b_0) = \sum_{\tau_{\mathcal{T}}} \mathbb{P}^{\pi}(\tau_{\mathcal{T}}) \left[\sum_{t=0}^{\mathcal{T}} r_x(x_t, a_t)\right] \tag{A.207}$$

applying chain rule and rearranging terms,

$$= \sum_{\tau_{\mathcal{T}}} \mathbb{P}^{\pi}(x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}} \mid \tau_0) \mathbb{P}^{\pi}(\tau_0) \left[\sum_{t=0}^{\mathcal{T}} r_x(x_t, a_t)\right] \tag{A.208}$$

$$= \sum_{\tau_0} \mathbb{P}^{\pi}(\tau_0) \sum_{x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}}} \mathbb{P}^{\pi}(x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}} \mid \tau_0) \left[\sum_{t=0}^{\mathcal{T}} r_x(x_t, a_t)\right] \tag{A.209}$$

$$= \sum_{\tau_0} \mathbb{P}^{\pi}(\tau_0) \left[r_x(x_0, a_0) + \sum_{x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}}} \mathbb{P}^{\pi}(x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}} \mid \tau_0) \left[\sum_{t=1}^{\mathcal{T}} r_x(x_t, a_t)\right]\right] \tag{A.210}$$

nullifying instances of the complete probability distribution, $\mathbb{P}^{\pi}(\cdot)$, is denoted as a simplified distribution, $\bar{\mathbb{P}}^{\pi}(\cdot)$. We can then split and bound from above the value function, such that the simplified value function consideres only a subset of the trajectories at time $t = 0$,

$$\leq \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \left[r_x(x_0, a_0) + \sum_{x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}}} \mathbb{P}^{\pi}(x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}} \mid \tau_0) \left[\sum_{t=1}^{\mathcal{T}} r_x(x_t, a_t) \right] \right] \tag{A.211}$$

$$+ \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \right] \mathcal{V}_{max,0} \tag{A.212}$$

We then apply similar steps on the next time step, $t = 1$,

$$= \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \right] \mathcal{V}_{max,0} + \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \Bigg[r_x(x_0, a_0) \tag{A.213}$$

$$+ \sum_{x_{1:\mathcal{T}}, z_{1:\mathcal{T}}, a_{1:\mathcal{T}}} \mathbb{P}^{\pi}(x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}} \mid \tau_1) \mathbb{P}^{\pi}(x_1, z_1, a_1 \mid \tau_0) \left[\sum_{t=1}^{\mathcal{T}} r_x(x_t, a_t) \right] \Bigg]$$

$$= \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \right] \mathcal{V}_{max,0} + \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \Bigg[r_x(x_0, a_0) \tag{A.214}$$

$$+ \sum_{x_1, z_1, a_1} \mathbb{P}^{\pi}(x_1, z_1, a_1 \mid \tau_0) \sum_{x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}}} \mathbb{P}^{\pi}(x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}} \mid \tau_1) \left[\sum_{t=1}^{\mathcal{T}} r_x(x_t, a_t) \right] \Bigg]$$

$$= \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \right] \mathcal{V}_{max,0} + \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \Bigg[r_x(x_0, a_0) \tag{A.215}$$

$$+ \sum_{x_1, z_1, a_1} \mathbb{P}^{\pi}(x_1, z_1, a_1 \mid \tau_0) \left[r_x(x_1, a_1) + \sum_{x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}}} \mathbb{P}^{\pi}(x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}} \mid \tau_1) \left[\sum_{t=2}^{\mathcal{T}} r_x(x_t, a_t) \right] \right] \Bigg]$$

$$\leq \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \left[1 - \sum_{x_1, z_1, a_1} \bar{\mathbb{P}}^{\pi}(x_1, z_1, a_1 \mid \tau_0) \right] \mathcal{V}_{max,1} + \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \right] \mathcal{V}_{max,0} \tag{A.216}$$

$$+ \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \Bigg[r_x(x_0, a_0) + \sum_{x_1, z_1, a_1} \bar{\mathbb{P}}^{\pi}(x_1, z_1, a_1 \mid \tau_0) \Bigg[r_x(x_1, a_1)$$

$$+ \sum_{x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}}} \mathbb{P}^{\pi}(x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}} \mid \tau_1) \left[\sum_{t=2}^{\mathcal{T}} r_x(x_t, a_t) \right] \Bigg] \Bigg]$$

which results in,

$$= \left[\sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) - \sum_{\tau_1} \bar{\mathbb{P}}^{\pi}(\tau_1) \right] \mathcal{V}_{max,1} + \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \right] \mathcal{V}_{max,0} \tag{A.217}$$

$$+ \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0) \Bigg[r_x(x_0, a_0) + \sum_{x_1, z_1, a_1} \bar{\mathbb{P}}^{\pi}(x_1, z_1, a_1 \mid x_0, a_0) \Bigg[r_x(x_1, a_1)$$

$$+ \sum_{x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}}} \mathbb{P}^{\pi}(x_{2:\mathcal{T}}, z_{2:\mathcal{T}}, a_{2:\mathcal{T}} \mid \tau_1) \left[\sum_{t=2}^{\mathcal{T}} r_x(x_t, a_t) \right] \Bigg] \Bigg]$$

Performing the same steps iteratively up to time $t = \mathcal{T}$, yields the desired outcome,

$$V^{\pi}(b_0) \leq \sum_{t=0}^{\mathcal{T}} \sum_{\tau_t} \bar{\mathbb{P}}^{\pi}(\tau_t) r_x(x_t, a_t) + \mathcal{V}_{max,0} \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^{\pi}(\tau_0)\right] + \sum_{t=0}^{\mathcal{T}-1} \mathcal{V}_{max,t+1} \left[\sum_{\tau_t} \bar{\mathbb{P}}^{\pi}(\tau_t) - \sum_{\tau_{t+1}} \bar{\mathbb{P}}^{\pi}(\tau_{t+1})\right] \tag{A.218}$$

A.6.5 Optimality Guarantees

Lemma A.6.3. *Let $\mathcal{A}$ be the set of actions and $\mathcal{U}_0^{\star}(H_t)$, $\mathcal{L}_0^{\star}(H_t)$ be the upper and lower bounds of node H_t chosen according to,*

$$\mathcal{U}_0^{\star}(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \left[r_x(x_t, a_t) + \mathcal{V}_{\max,t}\right] + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{U}_0^{\star}(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\max,t}\right] \tag{A.219}$$

$$\mathcal{L}_0^{\star}(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \left[r_x(x_t, a_t) + \mathcal{V}_{\min,t}\right] + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{L}_0^{\star}(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\min,t}\right] \tag{A.220}$$

and,

$$\mathcal{U}_0^{\star}(H_T) \triangleq \sum_{\tau_{\mathcal{T}} \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r_x(x_T), \qquad \mathcal{L}_0^{\star}(H_T) \triangleq \sum_{\tau_{\mathcal{T}} \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r_x(x_T). \tag{A.221}$$

Then, the optimal root-value is bounded by,

$$\mathcal{L}_0^{\star}(H_0) \leq V^{\pi^*}(H_0) \leq \mathcal{U}_0^{\star}(H_0). \tag{A.222}$$

Proof. We wish to show that $\mathcal{L}_0^{\star}(H_0) \leq V^{\pi^*}(b_0) \leq \mathcal{U}_0^{\star}(H_0)$. We derive a proof for one side of the inequality, while the other follows similarly. First note that,

$$V^{\pi^*}(b_0) \leq \mathcal{U}_0^{\pi^*}(H_0) \leq \max_{\pi \in \Pi} \mathcal{U}_0^{\pi}(H_0) \tag{A.223}$$

where the first inequality is due to Theorem 5.3, and the second inequality is true by definition. However, the claim in Lemma 5.3.2 is a recursive claim, while the bound provided in Theorem 5.3 only holds with respect to the root. Thus, for completeness, we also need to show that the best action can be chosen recursively, even though the bound is 'partial' in different parts of the tree.

$$
\begin{aligned}
&\max_{\pi_{0:\mathcal{T}}\in\Pi} \mathcal{U}_0^{\pi}(H_0) \\
&= \max_{\pi_{0:\mathcal{T}}\in\Pi} \sum_{\tau_0\in\mathcal{T}(H_0)} \overline{\mathbb{P}}(\tau_0)[r_x(x_0,\pi_0)+\mathcal{V}_{max,0}] + \sum_{z_1\in\overline{\mathcal{Z}}(H_0,\pi_0)} \left[\mathcal{U}_0^{\pi}(H_1) - \sum_{\tau_1\in\mathcal{T}(H_1)} \overline{\mathbb{P}}(\tau_1)\mathcal{V}_{max,0}\right] \\
&= \max_{\pi_0\in\Pi} \left\{ \sum_{\tau_0\in\mathcal{T}(H_0)} \overline{\mathbb{P}}(\tau_0)[r_x(x_0,\pi_0)+\mathcal{V}_{max,0}] + \max_{\pi_{1:\mathcal{T}}\in\Pi} \sum_{z_1\in\overline{\mathcal{Z}}(H_0,\pi_0)} \left[\mathcal{U}_0^{\pi}(H_1) - \sum_{\tau_1\in\mathcal{T}(H_1)} \overline{\mathbb{P}}(\tau_1)\mathcal{V}_{max,0}\right] \right\} \\
&= \max_{a_0} \left\{ \sum_{\tau_0\in\mathcal{T}(H_0)} \overline{\mathbb{P}}(\tau_0)[r_x(x_0,a_0)+\mathcal{V}_{max,0}] + \sum_{z_1\in\overline{\mathcal{Z}}(H_0,a_0)} \left[\max_{\pi_{1:\mathcal{T}}\in\Pi} \mathcal{U}_0^{\pi}(H_1) - \sum_{\tau_1\in\mathcal{T}(H_1)} \overline{\mathbb{P}}(\tau_1)\mathcal{V}_{max,0}\right] \right\}
\end{aligned}
$$

which continues similarly up to time $t=\mathcal{T}$, which completes the proof,

$$
V^{\pi^*}(b_0) \leq \mathcal{U}_0^{\pi^*}(H_0) \leq \max_{\pi\in\Pi} \mathcal{U}_0^{\pi}(H_0) = \mathcal{U}_0^{\star}(H_0). \tag{A.224}
$$

Lemma A.6.4. *Performing exploration based on* (5.29), (5.30) *and* (5.31) *ensures that the algorithm converges to the optimal value function within a finite number of planning iterations.*

Proof. Consider a given policy π. We claim that following the state and observation selection criteria in equations (5.30) and (5.31) will lead to visiting unexplored trajectories τ_T at every iteration unless all relevant trajectories have already been explored.

To show this, note that the upper bound $\mathcal{U}_0^{\star}((H_t,a_t,o_{t+1}))$ and the lower bound $\mathcal{L}_0^{\star}((H_t,a_t,o_{t+1}))$ will converge when the bound interval is zero, i.e.,

$$
\mathcal{U}_0^{\star}((H_t,a_t,o_{t+1})) - \mathcal{L}_0^{\star}((H_t,a_t,o_{t+1})) = 0. \tag{A.225}
$$

This convergence occurs when all future trajectories by following policy π from node $H_{t+1}=(H_t,a_t,o_{t+1})$ until the end of the horizon were explored,

$$
\begin{aligned}
&\mathcal{U}_0^{\pi}(H_{t+1}) - \mathcal{L}_0^{\pi}(H_{t+1}) = \\
&= \sum_{\tau_{t+1}\in\mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\max,t+1} + \sum_{z_{t+2}\in\bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \left[\mathcal{U}_0^{\pi}(H_{t+2}) - \sum_{\tau_{t+2}\in\mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2})\mathcal{V}_{\max,t+1}\right] \\
&- \left[\sum_{\tau_{t+1}\in\mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\min,t+1} + \sum_{z_{t+2}\in\bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \left[\mathcal{L}_0^{\pi}(H_{t+2}) - \sum_{\tau_{t+2}\in\mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2})\mathcal{V}_{\min,t+1}\right]\right] \\
&= \left[\sum_{\tau_{t+1}\in\mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) - \sum_{z_{t+2}\in\bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \sum_{\tau_{t+2}\in\mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2})\right](\mathcal{V}_{\max,t+1} - \mathcal{V}_{\min,t+1}) \\
&+ \sum_{z_{t+2}\in\bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} [\mathcal{U}_0^{\pi}(H_{t+2}) - \mathcal{L}_0^{\pi}(H_{t+2})]
\end{aligned}
$$

since $\forall t \in [0, T-1]\,, \mathcal{V}_{\max,t+1} - \mathcal{V}_{\min,t+1} \neq 0$, then $\mathcal{U}_0^{\pi}(H_{t+1}) - \mathcal{L}_0^{\pi}(H_{t+1}) = 0$ only if ,

$$\sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) - \sum_{z_{t+2} \in \bar{\mathcal{Z}}(H_{t+1}, \pi_{t+1})} \sum_{\tau_{t+2} \in \mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2}) = 0, \ \forall t \in [0, T-2]\,. \tag{A.226}$$

Thus, all the simplified probability terms in the policy tree converge to 1. Similarly, the probability gap,

$$1 - \sum_{\tau_T} \bar{\mathbb{P}}^{\star}(\tau_T \mid \tau_t, a_t, z_{t+1}, x) = 0 \tag{A.227}$$

only when all non-zero future trajectories with a prefix $(\tau_t, a_t, z_{t+1}, x)$ have been explored. Finally, we are left to show that selecting actions based on the criteria shown in (5.29), results in the optimal action upon convergence. Utilizing lemma 5.3.2, the proof follows similarly to the one shown in (A.8), which concludes our derivation. ■

A.7 Experiments

A.7.1 POMDP scenarios

We begin with a brief description of the Partially Observable Markov Decision Process (POMDP) scenarios implemented for the experiments. each scenario was bounded by a finite number of time steps used for every episode, where each action taken by the agent led to a decrement in the number of time steps left. After the allowable time steps ended, the simulation was reset to its initial state.

Tiger POMDP

The Tiger is a classic POMDP problem [22], involves an agent making decisions between two doors, one concealing a tiger and the other a reward. The agent needs to choose among three actions, either open each one of the doors or listen to receive an observation about the tiger position. In our experiments, the POMDP was limited horizon of 5 steps. The problem consists of 3 actions, 2 observations and 2 states.

Discrete Light Dark

Is an adaptation from [48]. In this setting the agent needs to travel on a 1D grid to reach a target location. The grid is divided into a dark region, which offers noisy observations, and a light region, which offers accurate localization observations. The agent receives a penalty for every step and a reward for reaching the target location. The key challenge is to balance between information gathering by traveling towards the light area, and moving towards the goal region.

Laser Tag POMDP

In the Laser Tag problem, [46], an agent has to navigate through a grid world, shoot and tag opponents by using a laser gun. The main goal is to tag as many opponents as possible within a given time frame. The grid is segmented into various sections that have varying visibility, characterized by obstacles that block the line of sight, and open areas. There are five possible actions, moving in four cardinal directions (North, South, East, West) and shooting the laser. The observation space cardinality is $|\mathcal{Z}| \approx 1.5 \times 10^6$, which is described as a discretized normal distribution and reflect the distance measured by the laser. The states reflect the agent's current position and the opponents' positions. The agent receives a reward for tagging an opponent and a penalty for every movement, encouraging the agent to make strategic moves and shots.

Baby POMDP

The Baby POMDP is a classic problem that represents the scenario of a baby and a caregiver. The agent, playing the role of the caregiver, needs to infer the baby's needs based on its state, which can be either crying or quiet. The states in this problem represent the baby's needs, which could be hunger, discomfort or no need. The agent has three actions to choose from: feeding, changing the diaper, or doing nothing. The observations are binary, either the baby is crying or not. The crying observation does not uniquely identify the baby's state, as the baby may cry due to hunger or discomfort, which makes this a partially observable problem. The agent receives a reward when it correctly addresses the baby's needs and a penalty when the wrong action is taken.

A.7.2 Hyperparameters

The hyperparameters for both DB-DESPOT and AR-DESPOT algorithms were selected through a grid search. We explored an array of parameters for AR-DESPOT, choosing the highest-performing configuration. Specifically, the hyperparameter K was varied across $\{10, 50, 500, 5000\}$, while λ was evaluated at $\{0, 0.01, 0.1\}$. Similarly, DB-POMCP and POMCP were examined three different values for the exploration-exploitation weight, $c = \{0.1, 1.0, 10.0\}$ multiplied by V_{max}, which denotes an upper bound for the value function.

For the initialization of the upper and lower bounds used by the algorithms, we used the maximal reward, multiplied by the remaining time steps of the episode, $\mathcal{R}_{\max} \cdot (\mathcal{T} - t - 1)$.

Bibliography

[1] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A pomdp extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems (NIPS)*, pages 64–72, 2010.

[2] Moran Barenboim and Vadim Indelman. Adaptive information belief space planning. In *31st International Joint Conference on Artificial Intelligence and 25th European Conference on Artificial Intelligence (IJCAI-ECAI)*, 2022.

[3] Moran Barenboim, Moshe Shienman, and Vadim Indelman. Monte carlo planning in hybrid belief pomdps. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):6827–6834, 2023.

[4] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal. Particle filter based entropy. In *2010 13th International Conference on Information Fusion*, pages 1–8, 2010.

[5] S. Bowman, N. Atanasov, K. Daniilidis, and G. Pappas. Probabilistic data association for semantic slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1722–1729. IEEE, 2017.

[6] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[7] Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *Intl. Joint Conf. on AI (IJCAI)*, pages 1346–1352. Citeseer, 1997.

[8] Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. Continuous upper confidence trees. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pages 433–445. Springer, 2011.

[9] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012. GTSAM.

[10] K. Doherty, D. Fourie, and J. Leonard. Multimodal semantic slam with probabilistic data association. In *2019 international conference on robotics and automation (ICRA)*, pages 2419–2425. IEEE, 2019.

[11] Kevin J Doherty, David P Baxter, Edward Schneeweiss, and John J Leonard. Probabilistic data association via mixture models for robust semantic slam. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1098–1104. IEEE, 2020.

[12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer-Verlag, New York, 2001.

[13] Maxim Egorov, Zachary N. Sunberg, Edward Balaban, Tim A. Wheeler, Jayesh K. Gupta, and Mykel J. Kochenderfer. POMDPs.jl: A framework for sequential decision making under uncertainty. *Journal of Machine Learning Research*, 18(26):1–5, 2017.

[14] Johannes Fischer and Omer Sahin Tas. Information particle filter tree: An online algorithm for pomdps with belief-based rewards on continuous domains. In *Intl. Conf. on Machine Learning (ICML)*, Vienna, Austria, 2020.

[15] D. Fourie, J. Leonard, and M. Kaess. A nonparametric belief solution to the bayes tree. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[16] Marcus Hoerger and Hanna Kurniawati. An on-line pomdp solver for continuous observation spaces. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 7643–7649. IEEE, 2021.

[17] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *Intl. J. of Robotics Research*, pages 1271–1287, 2014.

[18] M. Hsiao and M. Kaess. Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2019.

[19] Ming Hsiao, Joshua G Mangelson, Sudharshan Suresh, Christian Debrunner, and Michael Kaess. Aras: Ambiguity-aware robust active slam based on multi-hypothesis state and map estimations. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5037–5044. IEEE, 2020.

[20] Qiangqiang Huang, Can Pu, Dehann Fourie, Kasra Khosoussi, Jonathan P How, and John J Leonard. Nf-isam: Incremental smoothing and mapping via normalizing flows. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021.

[21] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015.

[22] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31(2):217–236, Feb 2012.

[24] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. volume 49, pages 193–208. Springer, 2002.

[25] Tom Kennedy. Monte carlo methods-a special topics course. *University of Arizona*, 2016.

[26] A. Kim and R. M. Eustice. Active visual SLAM for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 34(4-5):457–475, 2014.

[27] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[28] 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. *Intl. J. of Robotics Research*, 36(10):1088–1130, August 2017.

[29] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.

[30] Hanna Kurniawati and Vinay Yadav. An online pomdp solver for uncertainty planning in dynamic environment. In *Robotics Research*, pages 611–629. Springer, 2016.

[31] Pierre-Yves Lajoie, Siyi Hu, Giovanni Beltrame, and Luca Carlone. Modeling perceptual aliasing in slam via discrete-continuous graphical models. *IEEE Robotics and Automation Letters (RA-L)*, 2019.

[32] Michael H Lim, Tyler J Becker, Mykel J Kochenderfer, Claire J Tomlin, and Zachary N Sunberg. Generalized optimality guarantees for solving continuous observation pomdps through particle belief mdp approximation. *arXiv preprint arXiv:2210.05015*, 2022.

[33] Michael H. Lim, Claire Tomlin, and Zachary N. Sunberg. Sparse tree search optimality guarantees in pomdps with continuous observation spaces. In *Intl. Joint Conf. on AI (IJCAI)*, pages 4135–4142, 7 2020.

[34] Erik G Miller. A new class of entropy estimators for multi-dimensional densities. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 3, pages III–297. IEEE, 2003.

[35] Beipeng Mu, Shih-Yuan Liu, Liam Paull, John Leonard, and Jonathan How. Slam with objects using a nonparametric pose graph. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[36] Mustafa Mukadam, Jing Dong, Xinyan Yan, Frank Dellaert, and Byron Boots. Continuous-time gaussian process motion planning via probabilistic inference. *Intl. J. of Robotics Research*, 37(11):1319–1340, 2018.

[37] Rémi Munos. *From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning.* 2014.

[38] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *Intl. J. of Robotics Research*, 32(7):826–840, 2013.

[39] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

[40] S. Pathak, A. Thomas, and V. Indelman. A unified framework for data association aware robust belief space planning and perception. *Intl. J. of Robotics Research*, 32(2-3):287–315, 2018.

[41] R. Platt, R. Tedrake, L.P. Kaelbling, and T. Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, pages 587–593, Zaragoza, Spain, 2010.

[42] Aleksandr V Segal and Ian D Reid. Hybrid inference optimization for robust pose graph estimation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2675–2682. IEEE, 2014.

[43] M. Shienman and V. Indelman. D2a-bsp: Distilled data association belief space planning with performance guarantees under budget constraints. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022.

[44] M. Shienman and V. Indelman. Nonmyopic distilled data association belief space planning under budget constraints. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2022.

[45] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2164–2172, 2010.

[46] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. In *NIPS*, volume 13, pages 1772–1780, 2013.

[47] Matthijs TJ Spaan, Tiago S Veiga, and Pedro U Lima. Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, 29(6):1157–1185, 2015.

[48] Zachary Sunberg and Mykel Kochenderfer. Online algorithms for pomdps with continuous state, action, and observation spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, 2018.

[49] Ori Sztyglic and Vadim Indelman. Online pomdp planning via simplification. *arXiv preprint arXiv:2105.05296*, 2021.

[50] Ori Sztyglic, Andrey Zhitnikov, and Vadim Indelman. Simplified belief-dependent reward mcts planning with guaranteed tree consistency. *arXiv preprint arXiv:2105.14239*, 2021.

[51] V. Tchuiev, Y. Feldman, and V. Indelman. Data association aware semantic mapping and localization via a viewpoint-dependent classifier model. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[52] Vincent Thomas, Geremy Hutin, and Olivier Buffet. Monte carlo information-oriented planning. *arXiv preprint arXiv:2103.11345*, 2021.

[53] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.

[54] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.

[55] Thomas Walsh, Sergiu Goschin, and Michael Littman. Integrating sample-based planning and model-based reinforcement learning. In *Nat. Conf. on Artificial Intelligence (AAAI)*, volume 24, 2010.

[56] Chenyang Wu, Guoyu Yang, Zongzhang Zhang, Yang Yu, Dong Li, Wulong Liu, and Jianye Hao. Adaptive online packing-guided search for pomdps. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 34, pages 28419–28430. Curran Associates, Inc., 2021.

[57] Nan Ye, Adhiraj Somani, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. *JAIR*, 58:231–266, 2017.

לאחר מכן, יישמנו פישוט למרחב המצב בהקשר של POMDPs עם מרחב מצב היברידי, הכולל מצבים דיסקרטיים ורציפים. בעיות מסוג זה מגוונות, וכוללות בין השאר בעיות בהן מקור התצפית אינו ידוע. ללא פישוט זה, מספר האפשרויות הדיסקרטיות במרחב ההיברידי עשויות לגדול באופן מעריכי עם מספר צעדי התכנון, כך שמספר הצעדים העתידיים אליהם ניתן לתכנן מוגבל למספר קטן. כדי להתמודד עם קושי זה, פישטנו את מספר האפשרויות לתת-קבוצה, והצגנו שני פתרונות עיקריים, כאשר הראשון מבוסס דגימה ומובטח להתכנס לפתרון הנכון, ואילו הפתרון השני מבוסס הבטחות מתמטיות החוסמות את המרחק האפשרי בין הפתרון התיאורטי, שאינו ידוע לנו, לבין הפתרון המקורב המחושב בפועל. בשני הפתרונות הראינו שיפור ביצועים ביחס לאלגוריתמים הקיימים כיום.

לבסוף, הרחבנו את הגישה שלנו לפישוט מרחב המצב והתצפיות גם יחד, בעבור בעיות POMDP דיסקרטיות ומתן הבטחות דטרמיניסטיות. הראינו שבכל רגע נתון בזמן התכנון ניתן לקבל חסמים לא רק על המרחק בין הפתרון התיאורטי למקורב, אלא גם על המרחק בין הפתרון האופטימלי שאיננו ידוע לנו, לבין הפתרון המקורב. בנוסף, הראינו שניתן להוסיף את החסמים לאלגוריתמים הטובים ביותר הידועים כיום עם תוספת קטנה לקושי החישובי. לבסוף, הראנו ששימוש בחסמים הדטרמיניסטיים עשויים לשפר את ביצועי אלגוריתמים אלו.

תקציר

סוכנים אוטונומיים הפועלים בעולם האמיתי נתקלים לעיתים קרובות בחוסר ודאות ומקבלים החלטות על בסיס מידע חלקי. ניתן למסגר את אתגר קבלת ההחלטות תחת מבנה מתמטי של תהליכי החלטה מרקוביים שניתנים לצפייה חלקית (POMDPs) . מאחר והמרחב בו פועל הסוכן ניתן לצפייה באופן חלקי בלבד, על הסוכן לשערך את המצב הקיים. שיערוך זה נשען באופן מסורתי על תיאוריית בייס, שתוצאתו היא פילוג עדכני על מרחב המצבים האפשריים בו נמצא הסוכן.

בעוד ש-POMDPs מציעים מסגרת מתאימה לתכנון תחת חוסר ודאות, מציאת תוכנית אופטימלית עבור POMDP יכולה להיות אינטנסיבית חישובית והיא אפשרית רק עבור משימות פשוטות ביותר. בבעיות מסוג זה, מספר התרחישים העתידיים גדל מעריכית עם מספר צעדי התכנון. בפרט, מספר התרחישים העתידי מעריכי במספר התצפיות האפשריות כפול מספר הפעולות. יתרה מכך, בכל תרחיש עתידי אפשרי, נדרש לחשב את הפילוג העדכני על מרחב המצבים, שעשוי להיות גדול, מה שמכביד עוד יותר על מציאת מדיניות אופטימלית בזמן סביר. כתוצאה מקושי זה, בשני העשורים האחרונים היינו עדים לעליית אלגוריתמים מקורבים, כמו חיפוש עצים וגישות מבוססות דגימה, כפתרונות מובילים להתמודדות עם בעיות POMDP מורכבות יותר. על אף היעילות שלהם, אלגוריתמים אלו בדרך כלל מציעים רק הבטחות הסתברותיות או, במקרים מסוימים, ללא הבטחות פורמליות כלל.

במחקרנו, התמקדנו בטיפול במגבלות אלו על ידי פיתוח מגוון אלגוריתמים מפושטים עם הבטחות פורמליות ודטרמיניסטיות. כדי לייעל את סיבוכיות החישוב, האלגוריתמים המפושטים המוצעים במחקרנו פועלים על תתי-קבוצות מתוך מרחב המצב והתצפיות, ומבצעים חישובים חלקיים מתוך החישוב התיאורתי, תוך מתן הבטחות מתמטיות ויעילות חישובית בהשוואה לאלגוריתמים הלא מפושטים. לצורך כך, הגדרנו POMDP אלטרנטיבי, מפושט יותר, הכולל רק שבריר ממרחב המצב או התצפיות.

תחילה, התמקדנו בפישוט מרחב התצפיות לטובת ייעול חישוב פונקציית הפרס. המתקדנו בבעיות בהן פונקציית הפרס מוגדרת כפונקצייה על פילוג הסתברות ולא רק כתוחלת על הפילוג. באופן מפורט יותר, פונקציית הפרס בבעיה זו הוגדרה כסכום ממושקל של תוחלת על פרס מבוסס מצב, כפי שמוגדר ברוב בעיות POMDP ואנטרופיה על פילוג המצבים. על פי רוב, מטרת השילוב של אנטרופיה מבוססת מצב היא מציאת מדיניות פעולה שלא דואגת רק למקסם את התועלת, אלא בד בבד למזער את חוסר הודאות של הסוכן הפועל בסביבה זו. בעבודתנו זו הראנו הבטחות ביצועים דטרמיניסטיות ביחס לבעיה הלא מפושטת, ושיפור זמני חישוב עד פי ארבע בהשוואה לאלגוריתם דומה ללא פישוט מרחב התצפיות.

המחקר בוצע בהנחייתו של פרופסור חבר ואדים אינדלמן, בפקולטה למערכות אוטונומיות ורובוטיקה.

חלק מן התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי-עת במהלך תקופת מחקר הדוקטורט של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

Moran Barenboim and Vadim Indelman. Adaptive information belief space planning. In *31st International Joint Conference on Artificial Intelligence and 25th European Conference on Artificial Intelligence (IJCAI-ECAI)*, 2022.

Moran Barenboim and Vadim Indelman. Online pomdp planning with anytime deterministic guarantees. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Moran Barenboim and Vadim Indelman. Journal version of online pomdp planning with anytime deterministic guarantees. 2024. To be submitted.

Moran Barenboim, Idan Lev-Yehudi, and Vadim Indelman. Data association aware pomdp planning with hypothesis pruning performance guarantees. *IEEE Robotics and Automation Letters (RA-L)*, 10, 2023.

Moran Barenboim, Moshe Shienman, and Vadim Indelman. Monte carlo planning in hybrid belief pomdps. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):6827–6834, 2023.

Idan Lev-Yehudi, Moran Barenboim, and Vadim Indelman. Simplifying complex observation models in continuous pomdp planning with probabilistic guarantees and practice. In *38th AAAI Conference on Artificial Intelligence (AAAI-24)*, 2024.

מחבר חיבור זה מצהיר כי המחקר, כולל איסוף הנתונים, עיבודם והצגתם, התייחסות והשוואה למחקרים קודמים וכו', נעשה כולו בצורה ישרה, כמצופה ממחקר מדעי המבוצע לפי אמות המידה האתיות של העולם האקדמי. כמו כן, הדיווח על המחקר ותוצאותיו בחיבור זה נעשה בצורה ישרה ומלאה, לפי אותן אמות מידה.

תודות

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

אלגוריתמים מפושטים לבעיות תחת אי ודאות עם הבטחות ביצועים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

מורן ברנבוים

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
סיוון תשפ"ד חיפה יולי 2024

אלגוריתמים מפושטים לבעיות תחת אי ודאות עם הבטחות ביצועים

מורן ברנבוים