Open-loop POMDP Simplification and Safe Skipping of Replanning with Formal Performance Guarantees

Da Kong[1] and Vadim Indelman[2,3]

[1]Technion Autonomous Systems Program (TASP)
[2]Stephen B. Klein Faculty of Aerospace Engineering
[3]Faculty of Data and Decision Sciences
Technion - Israel Institute of Technology, Haifa 320000, Israel
da-kong@campus.technion.ac.il, vadim.indelman@technion.ac.il

Abstract. Partially Observable Markov Decision Processes (POMDPs) provide a principled mathematical framework for decision-making under uncertainty. However, the exact solution to POMDPs is computationally intractable. In this paper, we address the computational intractability by introducing a novel framework for adaptive open-loop simplification with formal performance guarantees. Our method adaptively interleaves open-loop and closed-loop planning via a topology-based belief tree, enabling a significant reduction in planning complexity. The key contribution lies in the derivation of efficiently computable bounds which provide formal guarantees and can be used to ensure that our simplification can identify the immediate optimal action of the original POMDP problem. Our framework therefore provides computationally tractable performance guarantees for macro-actions within POMDPs. Furthermore, we propose a novel framework for safely skipping replanning during execution, supported by theoretical guarantees on multi-step open-loop action sequences. To the best of our knowledge, this framework is the first to address skipping replanning with formal performance guarantees. Practical online solvers for our proposed simplification are developed, including a sampling-based solver and an anytime solver. Empirical results demonstrate substantial computational speedups while maintaining provable performance guarantees, advancing the tractability and efficiency of POMDP planning.

1 Introduction

Partially Observable Markov Decision Processes (POMDPs) constitute a fundamental mathematical framework for sequential decision-making under uncertainty. Despite the theoretical elegance and broad applicability across diverse domains, POMDPs suffer from significant computational intractability, known as the *curse of dimensionality* and the *curse of history*.

Extensive research has focused on approximation methods to improve POMDP tractability [15,12,22,19,4,17,2]. Open-loop planning has emerged as a promising approach, reducing the belief tree complexity by eliminating observation

branches [5,1,7]. Unlike closed-loop methods that adapt to observations, open-loop approaches execute predefined action sequences without gathering observations during execution, also known as macro-actions [5].

However, open-loop planning and macro-actions typically yield suboptimal solutions without any performance guarantees. Although prior work [5] offers bounds for macro-actions, these rely on the computationally intractable Value of Information (VoI), restricting their practical online use. This limitation exposes a fundamental gap in the existing literature: the absence of tractable performance guarantees for open-loop approximations relative to the original POMDP—a gap that this work aims to address.

Furthermore, conventional online POMDP planners replan after executing only the first action, even in the open-loop approximation setting [7]. Skipping replanning can be an effective strategy to simplify POMDP planning at the execution level. While recent learning-based approaches have begun to address this challenge [8], the establishment of formal performance guarantees for safely skipping replanning remains an open problem that we tackle in this work.

Specifically, in this work, we propose a novel framework for adaptive open-loop POMDP simplification at two levels: first, it enables POMDP planning simplification while maintaining performance guarantees that ensure the identification of the same optimal action at the root as the original POMDP; second, it enables safely skipping replanning with formal performance guarantees.

More specifically, at the planning level, we propose a novel method that adaptively introduces the open-loop mode to some belief nodes, excluding observations to simplify POMDP planning with performance guarantees. As illustrated in Fig. 1, the incorporation of open-loop planning can dramatically reduce the size of the belief tree, which directly corresponds to a reduction in planning complexity. Our method adaptively chains open-loop and closed-loop steps. We derive novel computationally tractable bounds that relate this simplified POMDP to the original POMDP. Importantly, our bounds only depend on the simplified problem, enabling POMDP simplification with computationally tractable performance guarantees. This, in turn, enables online adaptation between open-loop and closed-loop branches in the belief tree by utilizing the corresponding bounds. To the best of our knowledge, this constitutes the first work to provide computationally tractable formal guarantees for incorporating open-loop planning, and equivalently, macro-actions into POMDPs.

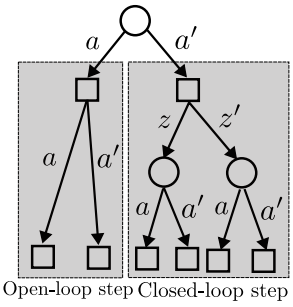

Fig. 1: A hybrid belief tree demonstrating the computational advantage of open-loop planning. The left branch employs open-loop action sequences, while the right branch utilizes traditional closed-loop planning.

At the execution level, we propose the first framework for skipping replanning in POMDPs with formal performance guarantees. Specifically, we establish novel

bounds on the posterior Q-values at future planning sessions based solely on the current planning session. We show that whenever these bounds satisfy certain conditions, the agent can safely skip replanning and execute multi-step open-loop actions that are guaranteed to be identical to the immediate optimal actions of the original POMDP problem, obtained through explicit replanning.

Further, we employ a sampling-based solver and an anytime solver for the derived bounds and demonstrate our method's effectiveness in simulated POMDP environments. Our approach achieves significant computational speedups while maintaining provable performance guarantees, highlighting its potential for real-world applications.

To summarize, our main contributions are as follows:

- We introduce the first POMDP simplification framework that adaptively incorporates open-loop steps and provides formal performance guarantees for both planning and execution.
- For planning, we develop novel efficiently computable bounds for introducing adaptive open-loop steps in POMDP planning that yield provable performance guarantees for our simplification method.
- For execution, we derive novel bounds for executing multi-step open-loop steps in POMDPs, providing guarantees for safely skipping replanning.
- We develop practical sampling-based and anytime MCTS-style solvers for our simplification framework, namely, AT-SparsePFT and AT-POMCP. They demonstrate substantial computational improvements through empirical evaluation while maintaining theoretical guarantees.

2 Related Work

For general background on POMDPs in robotics, we refer to surveys [13,14]. This section will focus on two specific aspects most relevant to this work: open-loop planning of POMDP and skipping replanning.

Open-Loop Planning. Practical POMDP solvers employ various approximation techniques to address the computational intractability, including approximate belief representations [23,15,19] and memory-based approximations [9,25,21]. Beyond approximation, simplification of POMDP with formal performance guarantees is essential for safety-critical tasks, including research on simplifying the observation model [17] and simplifying the state-observation space [2].

Within the POMDP literature, open-loop planning is often formulated as *macro-actions*. He et al. [7] established the first formal analysis of macro-action planning in POMDPs, providing crucial theoretical foundations for this approach. Subsequent work by Amato et al. [1] extended this framework to decentralized POMDPs, demonstrating the broader applicability of macro-action abstractions in multi-agent systems.

Recent theoretical advances include Flaspohler et al.'s [5] novel approach leveraging the Value of Information (VoI) to synthesize open-loop action sequences. While theoretically elegant, the substantial computational complexity of VoI calculations presents practical limitations for POMDP simplification.

From a robotic motion control perspective, Majumdar et al. [20] derived performance bounds for robot motion models that implicitly provide guarantees for open-loop planning in POMDPs, though these results do not directly address the POMDP simplification problem.

Contemporary research has shifted toward data-driven methods for macro-action sequence generation [16,18], demonstrating promising empirical results. However, these learning-based approaches currently lack the theoretical performance guarantees necessary for safety-critical applications.

Skipping Replanning. Conventional online planning pipelines employ a cyclical scheme of planning, execution, and replanning. However, the strategic decision of when to forgo replanning has received limited attention. Traditional approaches rely on hand-crafted replanning strategies, which often prove inflexible and sub-optimal. Honda et al. [8] recently addressed this gap through deep reinforcement learning, proposing adaptive replanning strategies for dynamic environments. However, their method lacks performance guarantees. To the best of our knowledge, we present the first work with theoretical analysis of replanning strategies in POMDPs with provable performance bounds.

3 POMDP Preliminaries

The basic model of POMDP is defined as a tuple: $\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathbb{P}_T, \mathbb{P}_Z, r, b_0 \rangle$, where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{Z}$ is the observation space. The transition model (or motion model) is defined as $\mathbb{P}_T(x_{k+1}|x_k, a_k)$, which describes the probabilistic transition of the state from $x_k \in \mathcal{X}$ to $x_{k+1} \in \mathcal{X}$ under a certain action $a_k \in \mathcal{A}$. The observation model is defined as $\mathbb{P}_Z(z_k|x_k)$, which describes the probability of observation $z_k \in \mathcal{Z}$ given a certain state $x_k \in \mathcal{X}$. The reward function r is considered to be state-dependent, $r(b, a) = \mathbb{E}_{x|b}[r(x, a)]$ and is bounded, i.e., $r \in [-R_{\max}, R_{\max}]$, with b representing the belief and a representing the action. b_0 is the initial belief.

Given that the true state is uncertain, a belief is maintained to represent the distribution of the current state given the history. The belief at any time instant k is defined as $b_k \triangleq P(x_k|h_k)$, where h_k is the history at that time, defined as $h_k \triangleq \{z_{1:k}, a_{0:k-1}\}$. A propagated history without the latest observation h_k^- is defined as $h_k^- = \{z_{1:k-1}, a_{0:k-1}\}$, and the corresponding propagated belief is $b_k^- \triangleq P(x_k|h_k^-)$.

A deterministic policy function is defined as $\pi : \mathcal{H} \mapsto \mathcal{A}$, which decides actions based on the history of beliefs. While a stochastic policy maps the history-action pairs to a probability as: $\pi : \mathcal{H} \times \mathcal{A} \mapsto [0, 1]$. The value function for a certain policy π over the planning horizon L is defined as the summation of all expected rewards: $V^\pi(b_k) = r(b_k, \pi_k(b_k)) + \sum_{i=k+1}^{k+L} \mathop{\mathbb{E}}_{z_{k+1:i}} \big[r(b_i, \pi_i(b_i))\big]$. The goal of a POMDP is to find the optimal policy π^* that maximizes the value function. The optimal policy can be calculated by the Bellman optimality as: $V^{\pi*}(b_k) = \max_{a_k} \Big[r(b_k, a_k) + \mathop{\mathbb{E}}_{z_{k+1}} V^{\pi*}(b_{k+1})\Big]$.

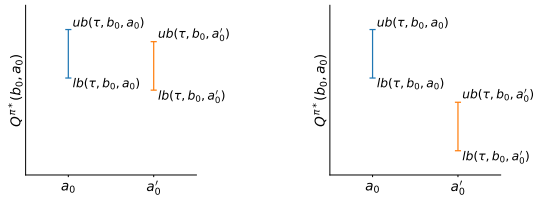

Fig. 2: Illustration of two cases for performance guarantees: (a) overlapping bounds necessitating topology refinement, and (b) non-overlapping bounds allowing for optimal action determination.

4 Open-loop as Simplification of POMDP

In this section, we present a novel framework for POMDP simplification that adaptively interleaves open-loop and closed-loop planning with formal performance guarantees. We derive computationally tractable bounds wherein both upper and lower bounds can be efficiently computed, depending solely on our simplification method and serving as the foundation for our performance guarantees. We denote the upper and lower bounds as ub and lb,

$$lb(\tau, b_0, a_0) \leq Q^{\pi *}(b_0, a_0) \leq ub(\tau, b_0, a_0), \tag{1}$$

where τ is a simplified topology, which can be seen as a control parameter of simplification of the POMDP problem and will be defined later in this section.

Such bounds can be used to determine the optimal action a_0 of the original POMDP problem in case there is no overlap between the highest lower bound and the second-highest upper bound. If there is overlap, we need to refine the topology τ to achieve non-overlapping bounds. Fig. 2 illustrates these two cases.

4.1 Definition

In POMDP, the policy space for both deterministic and stochastic policies is dependent on the history space. For the common full POMDP, the history will include the latest observation. In contrast, open-loop planning does not incorporate observations, leading to a simplified history space, which is the starting point of our simplification.

We define a history updater for *single-step* open-loop (OL) planning as $\psi^{OL}(h_k, a_k, z_{k+1}) = \{h_k, a_k\}$, for any possible observation $z_{k+1} \in \mathcal{Z}$, history $h_k \in \mathcal{H}_k$, and action $a_k \in \mathcal{A}$. This is in contrast to the standard *single-step* closed-loop (CL) history updater, which we denote as $\psi^{CL}(h_k, a_k, z_{k+1}) = \{h_k, a_k, z_{k+1}\}$. By iterating the single-step open-loop history updater, we can get the history for fully open-loop planning as: $h_k^{FOL} = \{a_{0:k-1}\}$. On the other hand, original POMDPs always use the closed-loop history updater, with a history of $h_k^{Full} = \{z_{1:k}, a_{0:k-1}\}$.

In order to incorporate open-loop and closed-loop steps adaptively, we introduce a topology τ to define the history and belief tree structure. We denote its history as Adaptive Open-Loop (AOL) history, $h_k^{\mathrm{AOL},\tau}$. The topology τ is defined as a set of binary indicator functions $\beta^{\mathrm{AOL},\tau}(h_k^{\mathrm{AOL},\tau})$. Specifically, $\beta^{\mathrm{AOL},\tau}(h_k^{\mathrm{AOL},\tau}) = 1$ indicates that a single-step open-loop planning is adopted for the history h_k^-, while $\beta^{\mathrm{AOL},\tau}(h_k^{\mathrm{AOL},\tau}) = 0$ indicates a single-step closed-loop planning. The *augmented* history update function ψ is defined as:

$$\begin{aligned} h_k^{\mathrm{AOL},\tau} &= \psi^{AOL}(\beta^{\mathrm{AOL},\tau}, h_{k-1}^{\mathrm{AOL},\tau}, a_{k-1}, z_k) \\ &= \begin{cases} \psi^{OL}(h_{k-1}^{\mathrm{AOL},\tau}, a_{k-1}, z_k), & \text{if } \beta^{\mathrm{AOL},\tau}(h_{k-1}^{\mathrm{AOL},\tau}) = 1, \\ \psi^{CL}(h_{k-1}^{\mathrm{AOL},\tau}, a_{k-1}, z_k), & \text{if } \beta^{\mathrm{AOL},\tau}(h_{k-1}^{\mathrm{AOL},\tau}) = 0. \end{cases} \end{aligned} \tag{2}$$

Here, $h_k^{\mathrm{AOL},\tau}$ denotes the augmented history at time step k under topology τ, which can be either open-loop or closed-loop depending on the indicator functions $\beta^{\mathrm{AOL},\tau}$. By recursively applying (2) from the initial belief to the planning horizon L, we construct a complete history space $\mathcal{H}_t^{\mathrm{AOL},\tau} = \{h_t^{\mathrm{AOL},\tau}\}$ and the corresponding belief tree $\mathbb{T}^{\mathrm{AOL},\tau} = \{\mathcal{H}_t^{\mathrm{AOL},\tau} : 1 \le t \le L\}$ for topology τ. This formulation encompasses two extreme cases: fully open-loop planning, where $\beta^{\mathrm{AOL},\tau} = 1$ for all nodes, and the original POMDP formulation with fully closed-loop planning, where $\beta^{\mathrm{AOL},\tau} = 0$ for all nodes.

This belief tree topology definition is an extension of [11], where the topology is used to switch between original and simplified observation spaces and models. In contrast, we use topology to chain open-loop and closed-loop steps, contributing to an adaptive simplification of history.

Further, the augmented policy is defined as $\pi_t^{\mathrm{AOL},\tau} : \mathcal{H}_t^{\mathrm{AOL},\tau} \mapsto \mathcal{A}$. It maps augmented histories to actions, enabling adaptive switching between open-loop action sequences and closed-loop policy steps according to the topology τ. The corresponding Q-function for adaptive policy is defined as $Q^{\pi^{\mathrm{AOL},\tau*}}(b_0, a_0) = \mathbb{E}_{x_0|b_0}[r(x_0, a_0) + \mathbb{E}_{z_1|b_0,a_0} \max_{a_1} Q^{\pi^{\mathrm{AOL},\tau*}}(b(h_1^{\mathrm{AOL},\tau}), a_1)]$. If $\beta^{\mathrm{AOL},\tau}(h_1^{\mathrm{AOL},\tau}) = 1$, as determined by the adaptive topology, the expectation over z_1 cancels out.

We now introduce another level of open-loop planning where the observations are not considered but full observability is assumed. We introduce the Adaptive Fully-Observable (AFO) policy $\pi^{\mathrm{AFO},\tau}$, which employs topology τ to control observability—assuming full observability without dependence on observations at designated nodes while maintaining partial observability at others. The corresponding adaptive history $h_t^{\mathrm{AFO},\tau}$ is updated according to:

$$\begin{aligned} h_t^{\mathrm{AFO},\tau} &= \psi^{\mathrm{AFO}}(\beta^{\mathrm{AFO},\tau}, h_{t-1}^{\mathrm{AFO},\tau}, a_{t-1}, z_t) \\ &= \begin{cases} h_{t-1}^{\mathrm{AFO},\tau} a_{t-1} x_t, & \text{if } \beta^{\mathrm{AFO},\tau}(h_{t-1}^{\mathrm{AFO},\tau}) = 1, \\ \psi^{CL}(h_{t-1}^{\mathrm{AFO},\tau}, a_{t-1}, z_t), & \text{if } \beta^{\mathrm{AFO},\tau}(h_{t-1}^{\mathrm{AFO},\tau}) = 0. \end{cases} \end{aligned} \tag{3}$$

Here, the indicator function $\beta^{\mathrm{AFO},\tau}(\cdot)$ within the topology τ determines the observability mode: $\beta^{\mathrm{AFO},\tau} = 1$ denotes full observability at the belief node, while $\beta^{\mathrm{AFO},\tau} = 0$ denotes partial observability. Similarly, the Q-function for the

AFO policy is defined as

$$Q^{\pi^{\mathrm{AFO},\tau*}}(b_0,a_0)=\mathbb{E}_{x_0|b_0}[r(x_0,a_0)+\mathbb{E}_{x_1|b_0,a_0}\mathbb{E}_{z_1|x_1}\max_{a_1}Q^{\pi^{\mathrm{AFO},\tau*}}(b(h_1^{\mathrm{AFO},\tau}),a_1)],$$

where if $\beta^{\mathrm{AFO},\tau}(h_1^{\mathrm{AFO},\tau})=1$, the expectation of z_1 will cancel out but the expectation of x_1 will remain since the true state is included in $h_1^{\mathrm{AFO},\tau}$.

4.2 Performance Guarantees

For the bounds in (1), we propose to adopt the Q-function of the optimal adaptive open-loop policy $\pi^{\mathrm{AOL},\tau*}$ for the lower bound lb, and the optimal adaptive fully-observable policy $\pi^{\mathrm{AFO},\tau*}$ for the upper bound ub. We now present our main result that establishes these upper and lower bounds over the optimal Q-function of the original POMDP problem.

Theorem 1. *Let π^* denote the optimal policy of the original POMDP. Consider some topology τ_U and τ_L, and denote the topology-dependent optimal augmented open-loop policy as π^{AOL,τ_L*}. In the same way, we denote the topology-dependent optimal adaptive fully-observable policy as π^{AFO,τ_U*}. Then,*

$$Q^{\pi^{AOL,\tau_L*}}(b_0,a_0)\le Q^{\pi^*}(b_0,a_0)\le Q^{\pi^{AFO,\tau_U*}}(b_0,a_0). \tag{4}$$

Proof. We provide the proof in Appendix I.

Since the topologies τ_U and τ_L are always specified together and share the same property, for simplicity, we denote $\tau=(\tau_U,\tau_L)$ and use τ as shorthand for this pair from now on when the meaning is unambiguous. Precisely, ub and lb may be induced by different (but jointly specified) topologies: the upper bound uses τ_U through the AFO policy, while the lower bound uses τ_L through the AOL policy. Since we always consider them as a bundle, we use a slight abuse of notation and use τ as shorthand from now on since the distinction is clear from context.

In particular, based on Theorem 1, the bounds in (1) are defined as

$$ub(\tau,b_0,a_0)\triangleq Q^{\pi^{\mathrm{AFO},\tau*}}(b_0,a_0),\quad lb(\tau,b_0,a_0)\triangleq Q^{\pi^{\mathrm{AOL},\tau*}}(b_0,a_0). \tag{5}$$

As long as the bounds are easier to compute than the original POMDP and have no overlap as shown in Fig. 2b, we can use them to determine the optimal action a_0 of the original POMDP problem and achieve the simplification of POMDP with formal performance guarantees.

When the bounds (1) for different actions overlap under topology τ, we have to explore an alternative topology τ' to achieve non-overlapping bounds through an iterative process, which continues until we identify the optimal action (see Fig. 2). The transition process typically switches some of the belief nodes from a simplified mode to the original POMDP mode. This iterative process monotonically tightens the bounds; moreover, the bounds converge to the optimal Q-function of the original POMDP after a finite number of iterations. We provide a detailed analysis of these properties in Appendix III.

4.3 Integration with Online POMDP Solvers

To bridge the gap between theoretical analysis and practical implementation, we propose to use online POMDP solvers to estimate the bounds for our proposed simplification method and adapt topology online. In this work we consider two such solvers: sparse-sampling-style solver, Sparse-PFT [19], and an anytime MCTS solver, namely POMCP [24], which estimates the bounds and adapts the topology in an anytime manner. Specifically, in this section we consider a particle belief POMDP setting, and introduce two solvers, AT-SparsePFT and AT-POMCP. The results could be extended to POMDPs with theoretical beliefs, i.e. belief-MDP, following [19]. Utilizing these solvers enables practical applications of our proposed adaptive open and closed loop online planning framework with formal guarantees.

AT-SparsePFT. The SparsePFT algorithm represents beliefs using weighted particles $\{x^i, w^i\}_{i=1}^N$, where N is the number of particles and w^i denotes the weight of the i-th particle. The belief is formally denoted as: $\bar{b}(x) \triangleq \frac{\sum_{i=1}^N w^i \delta(x^i - x)}{\sum_{i=1}^N w^i}$. Here $\delta(\cdot)$ is the Dirac delta function. Sparse-PFT uses sparse sampling [10] to construct a search tree that, for each posterior belief node in the tree, branches the entire action space and samples N^O observations. Each posterior belief node in the tree is updated using a particle filter.

To facilitate the bounds estimation in our simplification framework with adaptive topology, we extend SparsePFT to AT-SparsePFT (Adaptive Topology SparsePFT), which works on the belief tree with augmented history as defined in (2) and (3). Given a topology τ, AT-SparsePFT estimates the upper and lower bounds (4) over the optimal action-value function of the original (particle-belief) POMDP, with formal finite-time guarantees. This is possible since these bounds correspond to the optimal action-value function of the corresponding topology, considering AOL and AFO settings. We provide probabilistic guarantees on the estimation error of the estimated bounds by AT-SparsePFT for a given topology τ in the following theorem.

Theorem 2. *Fix an arbitrary $\lambda > 0$. Consider N^O being the observation sampling number, N being the state sampling number, and $C = \min\{N, N^O\}$. For every depth $d \in \{0, \ldots, L-1\}$ and every action $a_d \in A$. The following event holds with probability at least $1 - 2\,|A|\,(|A|C)^{\,L-d}\,\exp\!\Big(-\frac{C\lambda^2}{2V_{\max}^2}\Big)$:*

$$|\hat{ub}(\bar{b}_d, a_d, \tau) - ub(\bar{b}_d, a_d, \tau)| \leq t \quad , \quad |\hat{lb}(\bar{b}_d, a_d, \tau) - lb(\bar{b}_d, a_d, \tau)| \leq t, \tag{6}$$

where $t \triangleq \frac{(L-d)(L-d-1)}{2}\,\lambda$, and $\hat{ub}$ and $\hat{lb}$ are the estimated bounds by SparsePFT for a given topology τ as

$$\hat{ub}(\bar{b}_0, a_0, \tau) \triangleq \hat{Q}^{\pi^{AFO,\tau*}}(b_0, a_0) \quad , \quad \hat{lb}(\bar{b}_0, a_0, \tau) \triangleq \hat{Q}^{\pi^{AOL,\tau*}}(b_0, a_0). \tag{7}$$

Proof. We provide the proof in Appendix II.

We utilize these estimated bounds to identify the optimal action at the root belief node, same manner as shown in Fig. 2. Based on the probabilistic guarantee, we can get the confidence level for identifying the optimal action via the estimated bounds at the root belief node. In practice, we start from a highly-simplified initial topology, e.g., a completely open-loop setting, and utilize the AT-SparsePFT solver to estimate the bounds in (1) with probabilistic guarantees (Theorem 2). In case of an overlap at the root of the tree, as shown in Fig. 2a, we refine the topology and repeat the process until we can determine the optimal action via non-overlapping bounds, as shown in Fig. 2b. The topology transition process is described in Appendix IV. Notably, the AT-SparsePFT solver caches and reuses the belief nodes remaining unchanged when transitioning to the next topology, which significantly reduces the computational overhead.

However, the sparse-sampling-style solver is limited to a short planning horizon due to the exponential complexity on the action space. Next, we introduce an anytime solver that scales better to longer horizons.

AT-POMCP. We adapt the MCTS-style solver POMCP [24] to estimate the bounds used in the proposed simplification while adapting the topology in an anytime manner, namely AT-POMCP (Adaptive Topology POMCP). In AT-POMCP, the belief tree is constructed online according to the adaptive open-loop history under a given topology τ. During each simulation, when expanding new nodes, the solver uses the adaptive history updater defined in (2) or (3) to associate the new node with its corresponding history. Inspired by the progressive widening methodology [3] in continuous MCTS [26], we propose a similar mechanism to adapt online the topology in a progressive manner during the POMCP simulation. This means the topology will be updated once after a given number of iterations following the procedure described in Appendix IV. This budget of iterations is adaptively updated using progressing widening. A detailed algorithm is provided in Appendix V, with lines $15-22$ showing the progressive adaptation of topology. As far as we know, this is the first work to adapt the topology of a belief tree in an anytime progressive manner. We provide the following convergence theorem for AT-POMCP:

Theorem 3 (Convergence of AT-POMCP). *Considering a discrete POMDP, for a suitable choice of the UCB parameter c and progressive widening parameter of AT-POMCP, the value function estimated by AT-POMCP,* $\hat{V}^{\mathrm{AT}*}(b_0)$*, converges in probability to the optimal value function,* $V^*(b_0)$*:* $\hat{V}^{\mathrm{AT}*}(b_0) \xrightarrow{p} V^*(b_0)$.

Proof. The proof is provided in Appendix VI.

To summarize, in this section we proposed a methodology for adaptive open-loop simplification in POMDP planning with formal performance guarantees. Next, we leverage it to enable skipping replanning in POMDPs with formal guarantees.

5 Skipping Replanning with Performance Guarantees

Conventional online planners perform replanning at every time step after executing the optimal immediate action a_0^*, which was computed during the initial planning session at $t = 0$. In this section, we introduce a novel framework that enables skipping replanning at certain steps during execution while maintaining formal performance guarantees. We formalize this approach as *open-loop execution* of POMDPs with performance guarantees relative to standard online planners within the original POMDP framework, where agents replan after each execution step. This extends the *adaptive open-loop planning* presented in Section 4 to *open-loop execution*. To our knowledge, this is a novel methodology to address the fundamental question of "when to replan" [8].

The proposed framework for skipping replanning in POMDPs comprises two main phases: planning and execution. The planning phase employs the simplification method introduced in Section 4; the execution phase facilitates safe skipping of replanning. This framework is outlined in Algorithm 2 of Appendix VII.

The core idea underlying the decision to skip replanning at a *future* time instant k, during the *current* planning session at time instant $t = 0$, involves reasoning about and bounding the corresponding action-value function $Q^{\pi*}(b_k, a_k)$ for different future posterior beliefs b_k and actions a_k, with regard to the optimal policy sequence $\pi^*_{k+1:k+L}$. Conceptually, such bounds provide formal performance guarantees that, under suitable conditions, enable skipping replanning if the optimal immediate action a_k^* can be deterministically determined at time instant $t = 0$. As will be seen, performance guarantees must be evaluated sequentially for future time instants, as these guarantees are meaningful only when they hold for all preceding actions, allowing the agent to skip replanning in preceding sessions. In other words, the proposed framework can support skipping a replanning session at a future time instant k as long as it can be skipped also in all preceding time instances until then.

Our proposed framework for skipping replanning operates as an *execution-time* decision that fully utilizes the time allocated to action execution and observation collection. Conventional online planners must pause while the agent executes actions and gathers observations. In contrast, our method operates in parallel with execution, thus avoiding additional overhead when assessing the feasibility of skipping replanning. In scenarios where action execution demands significant time, this approach yields substantial simplifications not only at the planning level but also at the execution level. Compared to alternative methods that reason if replanning can be skipped at each present time, given the corresponding posterior belief that is conditioned on the history at that time, our approach fully exploits the execution time for action performance and observation acquisition, resulting in reduced replanning overhead, i.e., upon receiving an observation from the environment, the decision of whether to skip replanning is immediate.

The core of our proposed framework is to check the performance guarantees for actions a_k at future time steps with $k \in [1, L]$, as indicated in line 18 of Algorithm 2. For the Q-value of a_k and posterior belief b_k at the k-th future

planning session, Theorem 1 provides the bounds: $\mathrm{lb}(\tau, b_k, a_k) \leq Q^{\pi^*}(b_k, a_k) \leq \mathrm{ub}(\tau, b_k, a_k)$, where $Q^{\pi*}$ is the optimal Q-function at time k with an L-steps planning horizon. However, the posterior belief b_k is unknown at the current planning session (at time instant $t = 0$). Given a topology τ with open-loop actions in the first k steps and recursively assuming the performance guarantees hold for preceding actions $a_{1:k-1}$, we now reformulate these bounds for *any* future belief b_k to depend solely on information available at $t = 0$:

$$\mathrm{lb}^k(\tau, b_0, a_{0:k}) \leq \mathrm{lb}(\tau, b_k, a_k) \leq Q^{\pi^*}(b_k, a_k) \leq \mathrm{ub}(\tau, b_k, a_k) \leq \mathrm{ub}^k(\tau, b_0, a_{0:k}). \quad (8)$$

We can utilize these bounds, at time $t = 0$, to check overlap for different candidate future actions a_k and determine the optimal action a_k^*, using the same principle as shown in Fig. 2 and discussed in Section 4.2. We denote this process as checking *Skipping Replanning Guarantees* (SRG).

Specifically, consider a set of topologies $\mathcal{T}^k$, where each topology τ incorporates open-loop simplifications in the first k steps: $\mathcal{T}^k = \{\tau : \beta^\tau(h) = 1, \forall h \in \mathcal{H}_{0:k-1}\}$. We present the main theorem deriving the bounds in (8):

Theorem 4. *Consider the current time to be* 0 *and a topology* $\tau \in \mathcal{T}^k$*. Assuming a positive Q-function, we have:*

$$\mathrm{lb}^k(\tau, b_0, a_{0:k}) \triangleq C_k(\mathcal{Z}, \mathcal{X}^R) \left(\tilde{Q}_{L+k}^{\pi^{AOL,\tau*}}(b_0, a_{0:k-1}, a_k) - \sum_{i=0}^{k-1} \mathbb{E}[r(b_i, a_i)] \right), \quad (9)$$

$$\mathrm{ub}^k(\tau, b_0, a_{0:k}) \triangleq \frac{1}{C_k(\mathcal{Z}, \mathcal{X}^R)} \left(\tilde{Q}_{L+k}^{\pi^{AFO,\tau*}}(b_0, a_{0:k-1}, a_k) - \sum_{i=0}^{k-1} \mathbb{E}[r(b_i, a_i)] \right). \quad (10)$$

Here, $\tilde{Q}_{L+k}^{\pi^{AOL,\tau*}}(b_0, a_{0:k-1}, a_k)$ *is obtained by extending the planning horizon to* $L + k$*, enforcing the action sequence* $a_{0:k-1}$ *at the first* k *steps, and following the policy* π *afterwards, i.e.* $\tilde{Q}_{L+k}^{\pi^{\tau*}}(b_0, a_{0:k-1}, a_k) \triangleq \sum_{i=0}^{k} \mathop{\mathbb{E}}_{z_{1:i}} [r(b_i, a_i)] + \sum_{i=k+1}^{k+L} \mathop{\mathbb{E}}_{z_{1:i}} \left[r(b_i, \pi_i^{\tau*}(b_i)) \right]$*, and* $C_k(\mathcal{Z}, \mathcal{X}^R) \triangleq \prod_{j=1}^{k} \frac{\min_{z_j \in \mathcal{Z}, x_j \in \mathcal{X}^R} \{P(z_j|x_j) > 0\}}{\max_{z_j \in \mathcal{Z}, x_j \in \mathcal{X}^R} \{P(z_j|x_j) > 0\}}$*, with* $\mathcal{X}^R$ *being the reachable state space after executing* $a_{0:k-1}$ *from* b_0*, i.e.* $\mathcal{X}^R = \mathcal{X}_{\mathrm{reach}}(b_0, a_{0:k-1}) = \mathrm{supp}(\mathbb{P}(x_k|b_0, a_{0:k-1}))$*, and* $\sum_{i=0}^{k-1} \mathbb{E}[r(b_i, a_i)]$ *corresponds to the expected sum of rewards when executing the action sequence* $a_{0:k-1}$ *from* b_0*.*

Proof. We provide the proof in Appendix VIII.

We now formulate conditions for skipping replanning with optimality guarantees.

Proposition 1. *Consider the current time to be* $t = 0$*, the belief is* b_0*, and a topology* $\tau \in \mathcal{T}^k$*. For each* $i \in [1, k]$*, we can obtain* $\mathrm{lb}^i(\tau, b_0, a_{0:i})$ *and* $\mathrm{ub}^i(\tau, b_0, a_{0:i})$ *as in Theorem 4. If for each* $i \in [1, k]$*, we can identify the optimal action* a_i^**, i.e.,* $a_i^* = \arg\max_{a \in \mathcal{A}} \mathrm{ub}^i(\tau, b_0, (a_{0:i-1}^*, a))$ *and* $\mathrm{lb}^i(\tau, b_0, (a_{0:i-1}^*, a_i^*)) \geq \mathrm{ub}^i(\tau, b_0, (a_{0:i-1}^*, a_i')), \forall a_i' \neq a_i^*$*, as shown in Fig. 2b, then we can identify the optimal action sequence* $a_{0:k}^*$ *at time* $t = 0$ *for all the possible future beliefs. Thus, we can directly execute* $a_{0:k}^*$ *and replanning can be safely skipped for all future planning sessions between time instances* $[1, k]$*.*

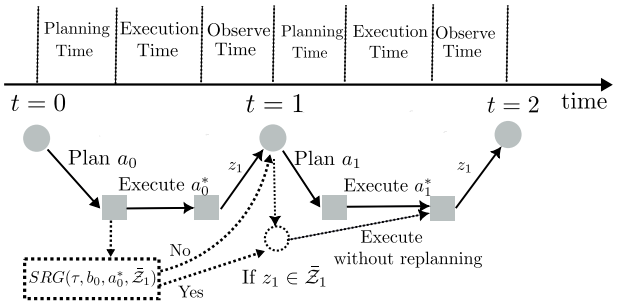

Fig. 3: The process of skipping replanning based on the allowed observation set $\bar{\mathcal{Z}}_1$. The agent executes the action a_0^* and checks if the observation z_1 belongs to the set $\bar{\mathcal{Z}}_1$. If it does, the agent skips replanning; otherwise, it triggers replanning. The process to skip replanning with performance guarantee is shown in dotted lines, which can be conducted in parallel with the execution of action and observation, thus reducing the replanning overhead.

However, the bounds presented in Theorem 4 may become uninformative in certain scenarios, as C_k accounts for all possible observations. To address this limitation, we now propose a refined variant that enables performance guarantee maintenance for skipping replanning across a broader range of scenarios.

We introduce a subset of observations $\bar{\mathcal{Z}}_{1:k}$, termed the *allowed observation set* for skipping replanning, where $\forall i \in [1,k], \quad \bar{\mathcal{Z}}_i \subseteq \mathcal{Z}$. We can tighten the bounds from Theorem 4 by constraining the factor C_k with $\bar{\mathcal{Z}}_{1:k}$, which can potentially restore the performance guarantees in some scenarios where the original formulation provides too loose bounds. For instance, in practice, the subset $\bar{\mathcal{Z}}_{1:k}$ can be constructed to include some of the most likely observations.

Given the allowed observation set $\bar{\mathcal{Z}}_{1:k}$, we can reformulate the bounds in (8) to account for this subset as $\bar{lb}^k(\tau, b_0, a_{0:k}, \bar{\mathcal{Z}}_{1:k})$ and $\bar{ub}^k(\tau, b_0, a_{0:k}, \bar{\mathcal{Z}}_{1:k})$, which leads to the following theorem:

Theorem 5. *Under the same conditions and definitions as in Theorem 4, but given a subset of observations $\bar{\mathcal{Z}}_{1:k} \subseteq \mathcal{Z}$, we have:*

$$\bar{lb}^k(\tau, b_0, a_{0:k}, \bar{\mathcal{Z}}_{1:k}) = C_k(\bar{\mathcal{Z}}_{1:k}, \mathcal{X}^R)\left(\tilde{Q}_{L+k}^{\pi^{AOL,\tau*}}(b_0, a_{0:k-1}, a_k) - \sum_{i=0}^{k-1}\mathbb{E}[r(b_i, a_i)]\right),$$

$$\bar{ub}^k(\tau, b_0, a_{0:k}, \bar{\mathcal{Z}}_{1:k}) = \frac{1}{C_k(\bar{\mathcal{Z}}_{1:k}, \mathcal{X}^R)}\left(\tilde{Q}_{L+k}^{\pi^{AFO,\tau*}}(b_0, a_{0:k-1}, a_k) - \sum_{i=0}^{k-1}\mathbb{E}[r(b_i, a_i)]\right),$$

where $C_k(\bar{\mathcal{Z}}_{1:k}, \mathcal{X}^R)$ is defined in Theorem 4.

Proof. We provide the proof in Appendix IX.

Based on the reformulated bounds, we can now determine whether the performance guarantee holds given a subspace $\bar{\mathcal{Z}}_{1:k}$ at time $t = 0$, i.e., if we can deter-

mine the optimal action a_k based on the bounds from Theorem 5 with respect to $\bar{\mathcal{Z}}_{1:k}$. We denote such a check by the Boolean function $SRG(\tau, b_0, a^*_{0:k-1}, a_k, \bar{\mathcal{Z}}_{1:k})$.

Although the bounds in the above theorems may be computationally expensive or scenario-dependent, they can be computed in parallel with the action execution and observation collection, thus avoiding additional overhead, which forms the *execution-time* planning. Fig. 3 illustrates this process of *execution-time* planning for the specific case when $k = 1$. The agent checks the SRG with $\bar{\mathcal{Z}}_1$ after the optimal action a^*_0 is identified. If SRG holds, i.e., $SRG(\tau, b_0, a^*_0, a_1, \bar{\mathcal{Z}}_1) =$ true, upon executing action a_0 and receiving observation z_1, the agent only needs to check if $z_1 \in \bar{\mathcal{Z}}_1$ to determine if replanning can be safely skipped. If $z_1 \in \bar{\mathcal{Z}}_1$, the agent can proceed executing a_1 without replanning; otherwise, replanning is triggered.

6 Experiments

We empirically evaluate our simplification methodology using practical solvers, AT-SparsePFT and AT-POMCP. Our experimental setup adopts the SparsePFT algorithm [19] and POMCP [24] as baselines. Results show that our approach preserves optimal decisions while substantially improving computational efficiency, yielding notable reductions in runtime without compromising quality.

We conduct the evaluation on the Beacon Navigation problem, where an agent is trying to reach a goal while avoiding obstacles under localization uncertainty. The robot receives noisy localization signals from beacons, reflecting real-world sensing conditions. This domain highlights the core challenge of planning under partial observability, requiring belief maintenance and informed action selection. It provides a practical yet tractable benchmark for evaluating belief space planning simplifications.

Detailed experimental settings are provided in Appendix X.

6.1 Open-loop Simplification for Planning

Bounds Estimation. We first present results from single-step simulation starting from a deterministic belief. Fig. 4 illustrates the estimated bounds obtained using our method compared with the Q-value estimated by standard sparse sampling. These results demonstrate that our proposed method effectively bounds the Q-value, thereby supporting our performance guarantees.

AT-SparsePFT Simulation Results. We conducted simulations over 10 steps for the beacon navigation problem, with each simulation repeated 100 times to compute means and standard deviations. Table 1 presents the results, comparing runtime and cumulative rewards after 10 steps. Our method, AT-SparsePFT, achieves a significant speedup of approximately 16× while maintaining comparable solution quality. The mean cumulative rewards obtained by our method closely match those of the standard SparsePFT approach, while providing substantial computational efficiency improvements. These results validate our claims regarding POMDP simplification with performance guarantees.

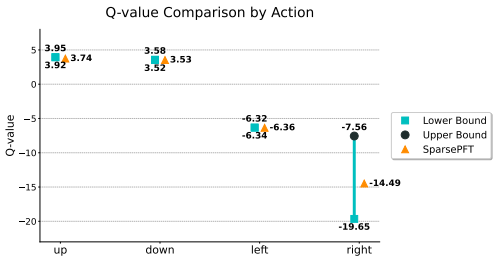

Fig. 4: Distribution of estimated bounds. The upper and lower bounds are computed using our proposed method with open-loop simplification, AT-SparsePFT. The yellow triangle denotes the Q-value estimated by the standard SparsePFT.

Method	Returns	Runtime (s)	Speedup Ratio
SparsePFT	12.64	345.9	1.0×
AT-SparsePFT (ours)	12.50	**20.66**	**16.7**×

Table 1: Cumulative rewards and runtime of the baseline SparsePFT method and our proposed method, AT-SparsePFT. The speedup ratio is computed as the ratio of baseline to proposed runtime. Results are averaged over 100 independent 10-step simulations.

AT-POMCP Simulation Results. We further conduct a comprehensive empirical evaluation of the proposed method estimated by the MCTS-style estimator (as introduced in Section 4.3) on the beacon navigation task over a planning horizon of 10 steps. Each experimental configuration is repeated 20 times to compute mean cumulative rewards and standard deviations. The computational time budget is progressively increased from 50 ms to 1 s for both the baseline POMCP method and our proposed approach. As to the progressive topology adaptation, the topology is set to be adapted every 100 simulations.

Table 2 presents a comparison of cumulative rewards achieved by the baseline POMCP method and our proposed anytime solver AT-POMCP under the same computational budgets. Our approach consistently achieves superior cumulative rewards across all tested time budgets, with improvements ranging from 6.84% to 10.24%. Given that the beacon navigation domain exhibits a sparse reward structure, these improvements in average cumulative rewards are noteworthy and statistically meaningful.

To assess the computational advantage of our approach, we compare the runtime required for the baseline POMCP algorithm and the proposed method when both achieve similar mean cumulative reward. Table 3 reports the experimental results: the baseline attains an average return of 7.456 after 2.5s of computation, whereas our method reaches a slightly higher return of 7.533 in only 0.05s. This corresponds to an approximately 50× speedup. The dramatic reduction in runtime while preserving reward quality validates our claim of the open-loop simplification of POMDP planning which yields substantial efficiency gains while preserving quality.

Time Budget	50ms	80ms	100ms	300ms	500ms	1.0s
POMCP	6.988 ± 0.352	7.082 ± 0.270	7.030 ± 0.165	7.14 ± 0.229	7.15 ± 0.284	7.347 ± 0.277
AT-POMCP(ours)	$\mathbf{7.533} \pm 0.481$	$\mathbf{7.566} \pm 0.533$	$\mathbf{7.698} \pm 0.494$	$\mathbf{7.825} \pm 0.655$	$\mathbf{7.883} \pm 0.353$	$\mathbf{8.018} \pm 1.526$
Improvement	7.80%	6.84%	9.52%	9.57%	10.24%	9.12%

Table 2: Cumulative reward comparison across computational budgets. Results are reported as mean ± standard deviation over 20 trials.

Method	Returns	Runtime (s)	Speedup Ratio
POMCP	7.456 ± 0.276	2.5	1.0×
AT-POMCP(ours)	7.533 ± 0.481	**0.05**	**50**×

Table 3: Runtime comparison between the baseline POMCP and the proposed AT-POMCP when operating at the same performance level (average cumulative reward). The speedup ratio is computed as the baseline runtime divided by the proposed method runtime; returns are averaged over 20 independent trials.

Table 3 demonstrates that our algorithm not only yields higher cumulative rewards but, more importantly, converges at a markedly faster rate than the baseline POMCP. The reward attained by the proposed method after merely 50 ms of computation would require roughly 2500 ms for the baseline to match—an approximately 50× acceleration in convergence. This pronounced speedup provides strong empirical evidence that the convergence rate of the simplified POMDP planner is dramatically superior, while simultaneously delivering equal solution quality. Consequently, the results support our theoretical analysis that our proposed open-loop simplification leads to improved planning efficiency in the online anytime solvers.

6.2 Experiments on Skipping Replanning

This section evaluates the effectiveness of our proposed framework in skipping replanning while preserving performance guarantees. We demonstrate this through a specific scenario, tunnel navigation, and show that skipping replanning is both feasible and safe. In this setup, similar to the beacon navigation problem, the agent navigates through a tunnel to reach the goal with noisy observations from beacons. We consider a positive reward function here. This scenario exemplifies a typical setting for open-loop planning, as previously explored in [6].

Fig. 5 illustrates the process of verifying the proposed framework of skipping replanning with formal guarantee. Using our proposed framework to safely skip replanning, at $t = 0$, the optimal action a_0^* can be identified and the future action a_1^* can be provably determined for all possible future observations $z_1 \in \bar{\mathcal{Z}}_1$, where $\bar{\mathcal{Z}}_1$ includes the four most likely observations.

Table 4 summarizes the results of our skipping replanning method. We compare AT-SparsePFT with and without our skipping replanning method. Our method successfully skips replanning in 24% of the steps while maintaining performance guarantees. The cumulative reward obtained by our method is almost identical to that of the method without skipping replanning due to the stochastic properties of the simulations, thereby validating the performance guarantees of

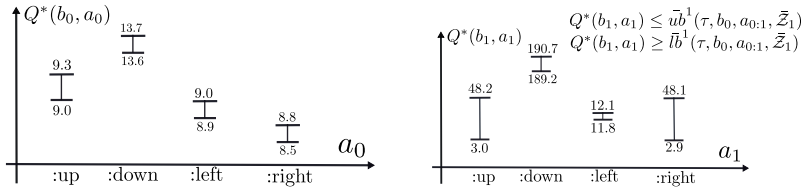

Fig. 5: Skipping replanning with guarantees: (a) Planning with the proposed adaptive open-loop simplification at $t = 0$. It shows $ub(\tau, b_0, a_0)$ and $lb(\tau, b_0, a_0)$ for different a_0. a_0^* is identified. (b) Bounding $Q^*(b_1, a_1)$ at $t = 0$ for any posterior belief b_1 that corresponds to future observations in $\bar{\mathcal{Z}}_1$. The figure shows $\bar{ub}^1$ and $\bar{lb}^1$ defined in Theorem 5. Here a_1^* can be provably determined at $t = 0$ for all realizations of the future observation z_1 in $\bar{\mathcal{Z}}_1$.

Method	Returns	Skipping Replanning Ratio
AT-SparsePFT w/o Skipping Replanning	235.7	0%
AT-SparsePFT w/ Skipping Replanning	231.3	**24**%

Table 4: Comparison of cumulative rewards and the ratio of steps where replanning can be safely skipped with performance guarantees. Results are averaged over 100 independent 5-step simulations.

our approach. The skipping ratio serves as a key metric, highlighting the efficiency gains of our framework at the overall execution level, in addition to the planning-level simplifications introduced in Section 4.

7 Conclusions

In this paper, we introduced a novel framework for adaptive open-loop simplification of POMDPs, enabling a significant reduction in computational complexity while maintaining formal performance guarantees. By leveraging a topology-based belief tree, our approach adaptively interleaves open-loop and closed-loop planning, providing efficiently computable bounds that guarantee identification of the optimal action in the original POMDP. We propose practical solvers for our adaptive simplification, AT-SparsePFT and AT-POMCP. Furthermore, we proposed a principled method for safely skipping replanning, supported by theoretical guarantees on multi-step open-loop action sequences. Empirical results demonstrate substantial speedup with provable guarantees, highlighting the practicality of our approaches.

Acknowledgments

This work was supported by the Israel Ministry of Innovation, Science and Technology.

References

1. Amato, C., Konidaris, G., Kaelbling, L.P., How, J.P.: Modeling and planning with macro-actions in decentralized pomdps. Journal of Artificial Intelligence Research **64**, 817–859 (2019)
2. Barenboim, M., Indelman, V.: Online pomdp planning with anytime deterministic optimality guarantees. Artificial Intelligence **350**, 104442 (2026). https://doi.org/https://doi.org/10.1016/j.artint.2025.104442
3. Couëtoux, A., Hoock, J.B., Sokolovska, N., Teytaud, O., Bonnard, N.: Continuous upper confidence trees. In: International conference on learning and intelligent optimization. pp. 433–445. Springer (2011)
4. Elimelech, K., Indelman, V.: Simplified decision making in the belief space using belief sparsification. Intl. J. of Robotics Research **41**(5), 470–496 (2022)
5. Flaspohler, G., Roy, N.A., Fisher III, J.W.: Belief-dependent macro-action discovery in pomdps using the value of information. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
6. Hauser, K.: Online planning in continuous pomdps with open-loop information-gathering plans. In: Intl. Conf. on Machine Learning (ICML) (2011)
7. He, R., Brunskill, E., Roy, N.: Efficient planning under uncertainty with macro-actions. J. of Artificial Intelligence Research pp. 523–570 (2011)
8. Honda, K., Yonetani, R., Nishimura, M., Kozuno, T.: When to replan? an adaptive replanning strategy for autonomous navigation using deep reinforcement learning. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 6650–6656. IEEE (2024)
9. Kara, A.D., Yuksel, S.: Near optimality of finite memory feedback policies in partially observed markov decision processes. J. of Machine Learning Research **23**(1), 437–482 (2022)
10. Kearns, M., Mansour, Y., Ng, A.Y.: A sparse sampling algorithm for near-optimal planning in large markov decision processes. Machine learning **49**(2), 193–208 (2002)
11. Kong, D., Indelman, V.: Simplified pomdp with an alternative observation space and formal performance guarantees. In: Proc. of the Intl. Symp. of Robotics Research (ISRR) (2024)
12. Kurniawati, H., Hsu, D., Lee, W.S.: SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: Robotics: Science and Systems (RSS) (2008)
13. Kurniawati, H.: Partially observable markov decision processes and robotics. Annual Review of Control, Robotics, and Autonomous Systems **5**(1), 253–277 (2022)
14. Lauri, M., Hsu, D., Pajarinen, J.: Partially observable markov decision processes in robotics: A survey. IEEE Transactions on Robotics **39**(1), 21–40 (2022)
15. Lee, W., Rong, N., Hsu, D.: What makes some pomdp problems easy to approximate? Advances in neural information processing systems **20** (2007)
16. Lee, Y., Cai, P., Hsu, D.: Magic: Learning macro-actions for online pomdp planning. In: Robotics: Science and Systems (RSS) (2021)
17. Lev-Yehudi, I., Barenboim, M., Indelman, V.: Simplifying complex observation models in continuous POMDP planning with probabilistic guarantees and practice. In: AAAI Conf. on Artificial Intelligence. vol. 38, pp. 20176–20184 (2024)
18. Liang, Y., Kurniawati, H.: Recurrent macro actions generator for pomdp planning. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2026–2033. IEEE (2023)

19. Lim, M.H., Becker, T.J., Kochenderfer, M.J., Tomlin, C.J., Sunberg, Z.N.: Optimality guarantees for particle belief approximation of pomdps. Journal of Artificial Intelligence Research **77**, 1591–1636 (2023)
20. Majumdar, A., Mei, Z., Pacelli, V.: Fundamental limits for sensor-based robot control. Intl. J. of Robotics Research (2023)
21. Patil, G., Mahajan, A., Precup, D.: On learning history-based policies for controlling markov decision processes. In: International Conference on Artificial Intelligence and Statistics. pp. 3511–3519. PMLR (2024)
22. Porta, J.M., Vlassis, N., Spaan, M.T., Poupart, P.: Point-based value iteration for continuous pomdps. J. of Machine Learning Research **7**, 2329–2367 (2006)
23. Roy, N., Gordon, G.J., Thrun, S.: Finding approximate pomdp solutions through belief compression. J. Artif. Intell. Res.(JAIR) **23**, 1–40 (2005)
24. Silver, D., Veness, J.: Monte-carlo planning in large pomdps. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2164–2172 (2010)
25. Subramanian, J., Mahajan, A.: Approximate information state for partially observed systems. In: IEEE Conference on Decision and Control. pp. 1629–1636. IEEE (2019)
26. Sunberg, Z., Kochenderfer, M.: Online algorithms for pomdps with continuous state, action, and observation spaces. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 28 (2018)