Generalized Optimality Guarantees for Solving Continuous Observation POMDPs through Particle Belief MDP Approximation

Michael H. Lim  
University of California, Berkeley, Department of Electrical Engineering and Computer Sciences, Berkeley, CA 94720 USA

Tyler J. Becker  
University of Colorado Boulder, Department of Aerospace Engineering Sciences, Boulder, CO 80309 USA

Mykel J. Kochenderfer  
Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305 USA

Clare J. Tomlin  
University of California, Berkeley, Department of Electrical Engineering and Computer Sciences, Berkeley, CA 94720 USA

Zachary N. Sunberg  
University of Colorado Boulder, Department of Aerospace Engineering Sciences, Boulder, CO 80309 USA

Abstract

Partially observable Markov decision processes (POMDPs) provide a flexible representation for real-world decision and control problems. However, POMDPs are notoriously difficult to solve, especially when the state and observation spaces are continuous or hybrid, which is often the case for physical systems. While recent online sampling-based POMDP algorithms that plan with observation likelihood weighting have shown practical effectiveness, a general theory bounding the approximation error of the particle filtering techniques that these algorithms use has not previously been proposed. Our main contribution is to formally justify that optimality guarantees in a finite sample particle belief MDP (PB-MDP) approximation of a POMDP/belief MDP yields optimality guarantees in the original POMDP as well. This fundamental bridge between PB-MDPs and POMDPs allows us to adapt any sampling-based MDP algorithm of choice to a POMDP by solving the corresponding particle belief MDP approximation and preserve the convergence guarantees in the POMDP. Practically, this means additionally assuming access to the observation density model, and simply swapping out the state transition generative model with a particle filtering-based model, which only increases the computational complexity by a factor of $O(C)$, with $C$ the number of particles in a particle belief state. In addition to our theoretical contribution, we perform five numerical experiments on benchmark POMDPs to demonstrate that a simple MDP algorithm adapted using PB-MDP approximation, Sparse-PFT, achieves performance competitive with other leading continuous observation POMDP solvers.

1. Introduction

Maintaining safety and acting efficiently in the midst of uncertainty is an important aspect in a diverse set of challenges from transportation (Holland et al., 2013; Sunberg & Kochenderfer, 2022) to autonomous scientific exploration (Bresina et al., 2002; Frew et al., 2020), to healthcare (Ayer et
al., 2012) and ecology (Memarzadeh & Boettiger, 2018). The partially observable Markov decision process (POMDP) is a flexible framework for sequential decision making in uncertain environments.

One common method for solving POMDPs is online tree search, which is attractive for several reasons. First, the approach scales to very large problems because it uses sampled trajectories, making it insensitive to the dimensionality of the state and observation spaces (Kearns et al., 2002). Second, since online computation focuses on the current states and states likely to be encountered in the future, it can reduce the need for offline computation and end-to-end training (Deglurkar et al., 2021). Third, tree search is applicable to a wide range of problems, for example hybrid continuous-discrete and nonconvex problems, because it only depends on a minimal set of problem structure requirements.

Recently proposed POMDP tree search algorithms (Garg et al., 2019; Lim et al., 2020, 2021; Mern et al., 2021; Sunberg & Kochenderfer, 2018; Wu et al., 2021) have been shown empirically to work on continuous state and observation spaces. Theoretical analysis, however, has lagged behind. While there are algorithms that have performance guarantees (Lim et al., 2020, 2021) and algorithms that perform well empirically (Garg et al., 2019; Lim et al., 2021; Mern et al., 2021; Sunberg & Kochenderfer, 2018; Wu et al., 2021), there has been little progress on generalization of why such family of algorithms can enjoy performance guarantees. For instance, AdaOPS (Wu et al., 2021), a recent particle belief-based POMDP solver that we include in our analysis, comes very close to capturing both theory and practice, but it has a complex algorithmic structure and provides only algorithm specific guarantees with additional simplifying assumptions. The considerable generalization gap between connecting POMDPs and practical approximations and solutions using particle methods still remains, only partially answered by a few algorithm-specific guarantees.

This manuscript formally justifies that optimality guarantees in a finite sample particle belief MDP (PB-MDP) approximation of a POMDP/belief MDP yields optimality guarantees in the original POMDP as well, by showing that the \( Q \)-values of the POMDP and PB-MDP are close with high probability by using an intermediary theoretical algorithm called Sparse Sampling-\( \omega \). Specifically, we prove that the Sparse Sampling-\( \omega \) \( Q \)-value estimates are close to both optimal \( Q \)-values of the POMDP and PB-MDP with high probability with our coupled convergence proof. This in turn implies that since there exists an algorithm that approximates both \( Q \)-values accurately with high probability, the optimal \( Q \)-values of the POMDP and PB-MDP themselves must be close to each other with high probability that scales as \( 1 - O(C^D \exp(-t \cdot C)) \).

This fundamental bridge between PB-MDPs and POMDPs allows us to adapt any sampling-based MDP algorithm of choice to a POMDP by solving the corresponding particle belief MDP approximation and preserve the convergence guarantees in the POMDP. Practically, this means additionally assuming we have an explicit observation model \( \mathcal{Z} \), and then simply swapping out the state transition generative model with a particle filtering-based model, which only increases the computational complexity by a factor of \( O(C) \), with \( C \) the number of particles in a particle belief state. This allows us to devise algorithms such as Sparse Particle Filter Tree (Sparse-PFT), which enjoys algorithmic simplicity, theoretical guarantees and practicality, as it is equivalent to upper confidence trees (UCT) (Bjarnason et al., 2009; Shah et al., 2022), with particle belief states.

The remainder of this paper proceeds as follows: First, Section 2 reviews preliminary definitions and previous related work. Section 3 formalizes the notion of particle belief MDPs (PB-MDPs) and gives detailed mathematical treatment. Then, Section 4 introduces Sparse Sampling-\( \omega \) algorithm, and proves its coupled convergence towards the optimal \( Q \)-values of a POMDP and its corresponding PB-MDP in Theorem 2. Section 5 formally bridges the gap between POMDPs and PB-MDPs
Approximate the POMDP as a Particle Belief MDP

Figure 1: Illustration of the proof of our main theorem, Theorem 3. Since Sparse Sampling-ω algorithm $Q$-value estimator converges to both the optimal $Q$-values of POMDP and PB-MDP, such an existence of algorithm implies that the optimal $Q$-values of POMDP and PB-MDP are also close to each other with high probability. This enables us to approximate the POMDP problem as a PB-MDP, and then solve the PB-MDP with an MDP algorithm to make a decision in the original POMDP while retaining the guarantees and computational efficiencies of the original MDP algorithm.

by leveraging the coupled convergence of Sparse Sampling-ω. In this section, we present two main theorems: Theorem 3 shows the optimal $Q$-value bounds between POMDP and PB-MDP, and Theorem 4 shows the near-optimality of planning in PB-MDP to solve a POMDP in closed-loop. We also introduce Sparse-PFT, a practical example of generating a PB-MDP approximation algorithm from an MDP algorithm. Finally, Section 6 empirically shows the performance of Sparse-PFT and other practical continuous observation POMDP algorithms over five different simulation experiments.

2. Background and Related Work

2.1 POMDPs

The partially observable Markov decision process (POMDP) is a mathematical formalism that can represent a wide range of sequential decision making problems (Kochenderfer, 2015). In a POMDP, an agent chooses actions based on observations to maximize the expectation of a cumulative reward signal. A POMDP is defined by the 7-tuple $(S, A, O, T, Z, R, \gamma)$. In this tuple, $S$, $A$, and $O$ are sets of all possible states, actions, and observations, respectively. These sets can be discrete, e.g. $\{1,2\}$, continuous, e.g. $\mathbb{R}^2$, or hybrid. The conditional probability distributions $T$ and $Z$ define state transitions and observation emissions, respectively. The transition probability distribution is conditioned on the current state $s$ and action $a$ and is denoted $T(s' \mid s, a)$. The observation probability is conditioned on the previous state and action and current state and denoted $Z(o \mid s, a, s')$. The reward function, $R(s, a)$, maps states and actions to an expected reward, and $\gamma \in [0, 1)$ is a discount factor. The agent plans starting from $b_0$, the initial state distribution or the initial belief. Some POMDP algorithms only require samples from the transition, observation, or reward models rather
than explicit knowledge of $T$, $Z$, or $R$. Such samples can be produced using a so-called generative model (Kearns et al., 2002) denoted with $s', o, r \leftarrow G(s, a)$.

The objective of a POMDP is to find an optimal policy, $\pi^*$, that selects actions that maximizes the discounted sum of future rewards, with an appropriate tie-breaking method:

$$\pi^* = \arg\max_\pi \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right].$$

(1)

In general, the actions may be chosen based on the entire history of actions and observations,

$$h_t \equiv (b_0, a_0, o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t).$$

(2)

However, because of the Markov property, it can be shown that optimal decisions can be made based only on the conditional distribution of the state given the history (Kaelbling et al., 1998), known as the belief,

$$b_t(s) \equiv \mathbb{P}(s_t = s \mid h_t).$$

(3)

This belief can be updated using Bayes’s rule or an efficient approximation such as a Kalman filter or particle filter, and it is often more straightforward to determine actions based on beliefs rather than the history. Since the belief and history fulfill the Markov property, a POMDP is a Markov decision process (MDP) on the belief or history space, commonly referred to as the belief MDP (Kaelbling et al., 1998).

In order to maximize the objective in Eq. (1), the policy must take into account the immediate reward from taking the action in the current state and whether the action will lead to states favorable for attaining rewards in the future. For a history $h$ and a corresponding belief $b$ history-action and belief-action value functions, defined as

$$Q^\pi(h, a) \equiv Q^\pi(b, a) \equiv \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid b_0 = b, a_0 = a, a_t = \pi(b_t) \right],$$

(4)

take both of these factors into account. When $\pi$ is also used for the current step, the expected accumulated reward from a history or belief is denoted with $V^\pi(h) = V^\pi(b) = Q^\pi(b, \pi(b))$. When $\pi$ is an optimal policy, these value functions are denoted with $Q^*$ and $V^*$. If $Q^*$ can be calculated, an optimal policy $\pi^*$ can simply be extracted with $\pi^*(h) = \arg\max_a Q^*(h, a)$. Thus, a common strategy for solving POMDPs involves iteratively improving estimates of $Q^*$ denoted simply with $Q$ for brevity.

Early research (Kaelbling et al., 1998; Shani et al., 2013; Smallwood & Sondik, 1973) sought to find optimal solutions to POMDPs offline; that is, they attempted to optimize actions for every possible belief before interacting with the environment. However, since POMDPs are generally intractable (Papadimitriou & Tsitsiklis, 1987), it is often impossible to find a complete solution for a POMDP offline. Instead, we seek to compute solutions online only for the part of the problem that may be reached in the immediate future.

### 2.2 Importance Sampling and Particle Filtering

In many real-world applications, updating the belief exactly based on a new action and observation is impractical. Fortunately, Monte Carlo methods provide simple and effective tools for approximate reasoning about distributions such as beliefs.
We often need to reason about a random variable $X \sim \mathcal{P}$ based only on samples from another related random variable, $Y \sim \mathcal{Q}$. Importance sampling allows us to, among other tasks, calculate the expectation of a function by observing that

$$
\mathbb{E}_{X \sim \mathcal{P}}[f(X)] = \int f(x)\mathcal{P}(x)dx = \int f(x)\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}\mathcal{Q}(x)dx \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\mathcal{P}(y_i)}{\mathcal{Q}(y_i)}f(y_i),
$$

where $\{y_i\}_{i=1}^{N}$ are samples from distribution $\mathcal{Q}$. The convergence property of this approximation relevant to the present work is described formally in Section 4.2.1.

The particle filter is an application of Monte Carlo estimation to the task of Bayesian belief updating (Kochenderfer & Wheeler, 2019; Thrun et al., 2005). The simplest form is an unweighted particle filter, in which the belief is represented by a collection of $N$ states, known as particles, $\hat{b} = \{(s_i)\}_{i=1}^{N}$. The density is approximated by $\hat{b}(s) \approx \sum_{i=1}^{N} \delta(s_i = s)$, where $\delta(\cdot)$ is a Dirac or Kronecker delta function depending on the form of the state space. At each step of the POMDP, after an action $a$ is taken and an observation $o$ is received, a new state $s'_i$ and observation $o_i$ is simulated once or more for each of the particles to create the new belief, $\hat{b}' = \{s'_i : o_i = o\}$. In most cases, few particles will match $o$ so it is difficult to maintain a large number of particles in the belief. Various domain-specific techniques can be used to reduce this problem, but it is difficult to solve completely in the unweighted particle filter.

The weighted particle filter is usually much more effective. The belief is represented by a collection of state particles and corresponding weights, $\hat{b} = \{(s_i, w_i)\}_{i=1}^{N}$. The density is approximated with $\hat{b}(s) \approx \frac{\sum_{i=1}^{N} w_i \delta(s_i = s)}{\sum_{i=1}^{N} w_i}$. A belief update consists of simulating each particle once or more and then calculating the new weight according to the importance sampling correction:

$$
w'_i = w_i \cdot Z(o | s, a, s'),
$$

where $N(h)$ and $N(h, a)$ are the number of times a history and action node have been visited, respectively. The first term encourages tree expansion towards parts of the history space that

\[\text{Typically, the weights of a few particles grow much larger than the others, so a resampling step creates many particles from those with large weights and eliminates those with very small weights (Kochenderfer & Wheeler, 2019; Thrun et al., 2005).}\]

2.3 Monte Carlo Tree Search (MCTS)

Monte Carlo tree search (MCTS) is a common solution technique for Games, MDPs, and POMDPs (Browne et al., 2012; Silver & Veness, 2010). In an MDP context, MCTS constructs a tree consisting of state and state-action nodes. In a POMDP context, each node corresponds to an action- or observation-terminated history node, estimating $Q(h, a)$ at each action-terminated history node. The most common variant is called partially observable upper confidence trees (PO-UCT) or partially observable Monte Carlo planning (POMCP)\footnote{Strictly speaking, POMCP also includes a specialized unweighted particle filter update that re-uses simulations from the planning step, but the term is often used informally as a synonym for PO-UCT.} (Silver & Veness, 2010) and repeats the following four steps to construct the tree:

1. Search. A simulation proceeds from the root node with action selected according to the UCB criterion,

$$Q(h, a) + c \sqrt{\frac{\log(N(h))}{N(h, a)}},$$

where $N(h)$ and $N(h, a)$ are the number of times a history and action node have been visited, respectively. The first term encourages tree expansion towards parts of the history space that
are likely to have high value, and the second term encourages exploration in parts of the space that have not been visited as often. The hyperparameter $c$ balances these two considerations.

Once an action has been chosen, the next state, reward, and observation are generated by sampling from the POMDP stochastic model.

2. **Expansion.** Eventually, the simulation reaches a leaf node and it is expanded by creating new action nodes.

3. **Estimation.** After the expansion step, the value at the history node is estimated, either with a domain-specific heuristic, or with a rollout simulation.

4. **Update.** The estimates of $Q(h,a)$ at each of the visited action nodes are updated with the discounted reward and backed up value estimates from descendant nodes.

This approach builds a tree asymmetrically favoring regions of the history and action spaces that are likely to be visited when the optimal policy is executed.

### 2.4 Online POMDP Solvers

In addition to the PO-UCT algorithm described above, there are several other approaches to solve POMDPs through online planning. Early solvers attempted to use exact Bayesian belief updates on discrete state spaces (Ross et al., 2008), however, these are much less scalable than PO-UCT. Two other popular solvers with scalability similar to UCT are determinized sparse partially observable trees (DESPOT) (Ye et al., 2017) and adaptive belief trees (ABT) (Kurniawati & Yadav, 2016). DESPOT uses a small number of determinized scenarios instead of independent random simulations to reduce variance and relies on heuristic tree search guided by upper and lower bounds rather than Monte Carlo tree search. ABT is designed to efficiently adapt to changes in the environment without discarding previous computation.

Since PO-UCT, DESPOT, and ABT all rely on an unweighted particle representation, weighted particle representations can be used with continuous spaces. Without such modification, they fail to find optimal policies in continuous observation spaces because the probability of generating the same observation twice, and hence creating beliefs with multiple particles, is zero (Lim et al., 2020; Sunberg & Kochenderfer, 2018). Partially observable Monte Carlo planning with observation widening (POMCPow) approaches the continuous observation challenge by introducing a weighted particle filter and the continuous action challenge with progressive widening (Sunberg & Kochenderfer, 2018). DESPOT-$\alpha$ (Garg et al., 2019) incorporates a similar weighting scheme and uses the $\alpha$-vector concept to generalize value estimates between sibling nodes. Adaptive online packing-guided search (AdaOPS) (Wu et al., 2021) fuses similar observation branches in the search tree to improve performance. Lazy Belief Extraction for Continuous Observation POMDPs (LABECOP) (Hoerger & Kurniawati, 2021) builds the planning tree by re-weighting particles and extracting belief sequence values efficiently.

### 3. Particle Belief MDPs (PB-MDPs)

In this section, we define the corresponding particle belief MDP (PB-MDP) for a given POMDP. Deriving the corresponding particle belief MDP of a POMDP is equivalent to approximating the belief MDP with a finite number of particles.

**Definition 1** (Particle Belief MDP). The corresponding particle belief MDP for a given POMDP problem $P = (S,A,O,T,Z,R,\gamma)$ is the MDP $M_P = (\Sigma,A,\tau,\rho,\gamma)$ defined by the following elements:
\[ \textbf{\( \Sigma \): State space over particle beliefs \( \bar{b}_d \). An element in this set, \( \bar{b}_d \in \Sigma \), is a particle collection, \( \bar{b}_d = \{(s_{d,i}, w_{d,i})\}_{i=1}^C \), where \( s_{d,i} \in S, w_{d,i} \in \mathbb{R}_+^{+} \). Note that this space is not permutation invariant over the particles, meaning that particles of different orders are considered different elements in \( \Sigma \).} \]

\[ \textbf{\( A \): Action space. Remains the same as the original action space.} \]

\[ \textbf{\( \tau \): Transition density \( \tau(\bar{b}_{d+1} | \bar{b}_d, a) \): We define the likelihood weights \( w_{d,i} \) of particles \( s_{d,i} \) to be updated through unnormalized Bayes rule:} \]

\[ w_{d+1,i} = w_{d,i} \cdot Z(o | a, s_{d+1,i}). \tag{7} \]

Then, the transition probability from \( \bar{b}_d \) to \( \bar{b}_{d+1} \) by taking the action \( a \) can be defined as:

\[ \tau(\bar{b}_{d+1} | \bar{b}_d, a) \equiv \int_{O} P(\bar{b}_{d+1} | \bar{b}_d, a, o)P(o | \bar{b}_d, a)do. \tag{8} \]

The first term in the integrand product \( P(\bar{b}_{d+1} | \bar{b}_d, a, o) \) is the conditional transition density given some observation \( o \). Note that the likelihood weight Bayesian update step is deterministic given \( s_{d,i}, s_{d+1,i}, a \) and \( o \). Combining with the facts that the state transition density update for each particle is independent and that the case when \( \bar{b}_{d+1} = \{s'_{j}, w'_{j}\} \) results in the only nonzero integrand, such likelihood calculation simplifies into the product of the transition densities for each \( i \)-th index particle if such transition is valid:

\[ P(\bar{b}_{d+1} | \bar{b}_d, a, o) = \int_{\Sigma} P(\bar{b}_{d+1} | \bar{b}_d, a, o, \{s'_{j}, w'_{j}\})P(\{s'_{j}, w'_{j}\} | \bar{b}_d, a, o)d\sigma \tag{9} \]

\[ = \int_{\Sigma} \delta[\bar{b}_{d+1} = \{s'_{j}, w'_{j}\}] \prod_{i=1}^{C} T(s'_{i} | s_{d,i}, a)d\sigma \tag{10} \]

\[ = \begin{cases} \prod_{i=1}^{C} T(s_{d+1,i} | s_{d,i}, a) & \text{if } w_{d+1,i} = w_{d,i} \cdot Z(o | a, s_{d+1,i}) \forall i \\ 0 & \text{otherwise.} \end{cases} \tag{11} \]

The second term in the integrand product \( P(o | \bar{b}_d, a) \) is the observation likelihood given a particle belief and an action. This can be shown to be equivalent to weighted sum of observation likelihoods conditioning on the observation having been generated from the respective \( i \)-th particle:

\[ P(o | \bar{b}_d, a) = P(o | \{s_{d,i}, w_{d,i}\}, a) = \frac{\sum_{i=1}^{C} w_{d,i} \cdot P(o | s_{d,i}, a)}{\sum_{i=1}^{C} w_{d,i}} \tag{12} \]

\[ = \frac{\sum_{i=1}^{C} w_{d,i} \cdot \int_{s} Z(o | a, s')T(s' | s_{d,i}, a)ds'}{\sum_{i=1}^{C} w_{d,i}}. \tag{13} \]

Note that this density \( \tau \) is usually impossible or very difficult to calculate explicitly. However, it is rather easy to sample from it using generative models.

\[ \text{2. The } d \text{ subscript, the number of steps, is included for subscript order consistency with the rest of the paper, but is not meaningful in this context.} \]
• \( \rho \): Reward function \( \rho(\bar{b}_d, a) \):

\[
\rho(\bar{b}_d, a) = \frac{\sum_i w_{d,i} \cdot R(s_{d,i}, a)}{\sum_i w_{d,i}}.
\]

Note that if \( R \) is bounded by \( R_{\text{max}} \), \( \rho \) is also bounded with \( ||\rho||_\infty \leq R_{\text{max}} \), since the normalized weights sum to 1.

• \( \gamma \): Discount factor. Remains the same as the original discount factor.

The significance of defining a corresponding particle belief MDP is that we can directly adapt any sampling-based MDP algorithms to approximately solve a POMDP by only changing the transition generative model. The transition generative model will now be a sampler based on particle filtering, as the particle belief MDP deals with particle belief states. Furthermore, this allows \( Q \)-value convergence guarantees of the MDP algorithms to translate nicely into solving the POMDP, as we will prove later in this paper that the optimal \( Q \)-values of the POMDP \( Q^*_P \) and PB-MDP \( Q^*_M \) are close with high probability.

4. Sparse Sampling-\( \omega \)

In order to show that the optimal \( Q \)-values of the POMDP \( Q^*_P \) and PB-MDP \( Q^*_M \) are approximately equivalent, we first introduce an algorithm called Sparse Sampling-\( \omega \) (sparse sampling with weights), which will serve as a theoretical bridge between POMDP and PB-MDP. Sparse Sampling-\( \omega \) is a sparse sampling solver that uses particle belief states with particle likelihood weighting to deal with observation uncertainty. As is evident from the name, Sparse Sampling-\( \omega \) takes inspiration from sparse sampling (Kearns et al., 2002) for continuous state MDPs, using particle belief states. This algorithm can also be seen as a slight modification of POWSS (Lim et al., 2020), the first known algorithm to enjoy convergence guarantees to the optimal policy for continuous observation POMDP problems. This duality effectively lets us bridge POMDPs and PB-MDPs. Note that Sparse Sampling-\( \omega \) is purely a theoretical intermediary tool to bridge POMDPs and PB-MDPs, and fully expanding the state and action nodes is extremely computationally inefficient. Rather, the existence of such theoretically well-behaved algorithm is what lets us effectively bridge the gap between the optimal \( Q \)-values of the POMDP \( Q^*_P \) and PB-MDP \( Q^*_M \).

4.1 Algorithm Definition

First, we define the auxiliary functions for Sparse Sampling-\( \omega \) in Algorithm 1, which are also applicable to any sampling-based particle belief MDP planner. \( \text{GENPF} \) is the helper function to generate the next-step particle belief set, where the particles are evolved through the transition density \( T \) and the weights are updated through the observation density \( Z \). In \( \text{GENPF} \), the sampled states \( s_{i}' \) are inserted into each next-step particle belief set \( \bar{b}a_o a_j \) with the new weights \( w_{i}' = w_{i} \cdot Z(o_j | a, s_{i}') \), which are the adjusted probability of hypothetically sampling observation \( o_j \) from state \( s_{i}' \). Furthermore, the reward returned by \( \text{GENPF} \) is the particle likelihood weighted reward \( \rho = \sum_i w_{i}r_{i}/\sum_i w_{i} \) of the current particle belief state, which is a constant output for a fixed pair of \( \bar{b}, a \). The \( \text{PLAN} \) function is the entry point to the Sparse-PFT algorithm, which initializes the particle belief state, subsequently calls \( \text{SIMULATE} \) to build the tree and returns the best action with the highest \( Q \)-value.
We define the main planning functions in Sparse Sampling-\(\omega\), ESTIMATEV and ESTIMATEQ functions, in Algorithm[2]. The global variables are the discount factor \(\gamma\), the generative model \(G\), the observation width \(C\), and the planning depth \(D\). We use particle belief set \(\bar{b}\) at every step \(d\), which contain pairs \((s_i, w_i)\) that correspond to the generated sample and its corresponding weight. ESTIMATEV is a subroutine that returns the value function \(V\), for an estimated state or belief, by calling ESTIMATEQ for each action and returning the maximum. Similarly, ESTIMATEQ performs sampling and recursive calls to ESTIMATEV to estimate the \(Q\)-function at a given step with a weighted average. In ESTIMATEQ, Sparse Sampling-\(\omega\) samples the next particle belief state using GENPF.

Consequently, the Sparse Sampling-\(\omega\) policy action can be obtained by calling the value estimation function ESTIMATEV\((\bar{b}_0, 0)\) at the root node and taking an action that maximizes the \(Q\)-value. The particle belief set is initialized by drawing samples from \(b_0\) and setting weights to \(1/C\), as the samples were drawn directly from \(b_0\). Sparse Sampling-\(\omega\) is not computationally efficient as it fully expands the sparsely sampled tree with full particle belief states, and serves only to demonstrates theoretical convergence and is only practically applicable to very small toy POMDP problems.

Sparse Sampling-\(\omega\) is identical to the sparse sampling algorithm (Kearns et al., 2002) planning on a particle belief MDP. It also is a slight modification of the previously-published POWSS al-
algorithm (Lim et al., 2020). Specifically, whereas POWSS generates exactly one observation and corresponding new belief for each particle in a belief, Sparse Sampling-$\omega$ randomly selects a state to generate the observation each time GENPF is called in Line 2 of Algorithm 2. This means that Sparse Sampling-$\omega$ performs a Monte Carlo sampling estimate of the next step value, while POWSS performs an importance weighted summation over the estimates.

Most importantly, this duality of being a modification of POWSS algorithm maintaining similar convergence guarantees for POMDPs while simultaneously being an adaptation of the sparse sampling algorithm for particle belief MDP makes it the ideal candidate to bridge POMDPs and PB-MDPs together. As an added benefit, the definition of Sparse Sampling-$\omega$ is much simpler than the original POWSS algorithm, while still allowing us to use similar analysis techniques used in both POWSS and sparse sampling.

4.2 Theoretical Analysis

In this section, we will prove that Sparse Sampling-$\omega$ algorithm can be made to approximate both optimal $Q$-values of the POMDP $Q_P^*$ and PB-MDP $Q_{MP}^*$ arbitrarily close by increasing the observation width $C$. Theorem 2 proves that the Sparse Sampling-$\omega$ algorithm approximates these $Q$-values with high probability by combining results from self-normalized importance sampling estimators and POWSS optimality proofs (Lim et al., 2020) to prove the optimality in $Q_P^*$, and sparse sampling proof (Kearns et al., 2002) to prove the optimality in $Q_{MP}^*$.

4.2.1 Importance Sampling

We begin the theoretical portion of this work by stating an important property about self-normalized importance sampling estimators (SN estimators). We have previously published this property (Lim et al., 2020) but present it again here because of its importance to our analysis. One goal of importance sampling is to estimate an expected value of a function $f(x)$ where $x$ is drawn from a distribution $P$ while the estimator only has access to another distribution $Q$ along with the importance weights $w_{P/Q}(x) \propto P(x)/Q(x)$. This technique is crucial for Sparse Sampling-$\omega$ because we wish to estimate the value for beliefs conditioned on observation sequences while only being able to sample from the marginal distribution of states for a given action sequence.

We define the following quantities:

$$\tilde{w}_{P/Q}(x) \equiv \frac{w_{P/Q}(x)}{\sum_{i=1}^{N} w_{P/Q}(x_i)} \quad \text{(SN Importance Weight)}$$

$$d_\alpha(P||Q) \equiv \mathbb{E}_{x \sim Q}[w_{P/Q}(x)^\alpha] \quad \text{(Rényi Divergence)}$$

$$\tilde{\mu}_{P/Q} \equiv \sum_{i=1}^{N} \tilde{w}_{P/Q}(x_i)f(x_i). \quad \text{(SN Estimator)}$$

Of particular importance is the infinite Rényi Divergence, $d_\infty$, which can be rewritten as an almost sure bound on the ratio of $P$ and $Q$:

$$d_\infty(P||Q) = \text{ess sup}_{x \sim Q} w_{P/Q}(x). \quad (15)$$

By assuming this bound is finite, we prove an estimator concentration bound in the following theorem.
Theorem 1 (SN $d_\infty$-Concentration Bound). Let $\mathcal{P}$ and $\mathcal{Q}$ be two probability measures on the measurable space $(\mathcal{X}, \mathcal{F})$ with $\mathcal{P}$ absolutely continuous w.r.t. $\mathcal{Q}$ and $d_\infty(\mathcal{P} \| \mathcal{Q}) < +\infty$. Let $x_1, \ldots, x_N$ be independent identically distributed random variables (i.i.d.r.v.) with distribution $\mathcal{Q}$, and $f : \mathcal{X} \rightarrow \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $\lambda > 0$ and $N$ large enough such that $\lambda > \|f\|_\infty d_\infty(\mathcal{P} \| \mathcal{Q}) / \sqrt{N}$, the following bound holds with probability at least $1 - 3 \exp(-N \cdot t^2(\lambda, N))$:

$$|\mathbb{E}_{x \sim \mathcal{P}}[f(x)] - \bar{\mu}_{\mathcal{P} \| \mathcal{Q}]| \leq \lambda,$$  

(16)

$$t(\lambda, N) \equiv \frac{\lambda \|f\|_\infty d_\infty(\mathcal{P} \| \mathcal{Q})}{1 / \sqrt{N}} - 1.$$  

(17)

Theorem 1 builds upon the derivation in Proposition D.3 of Metelli et al. (2018), which provides a polynomially decaying bound by assuming $d_2$ is bounded. Here, we compromise by further assuming that $d_\infty$ exists and is bounded to get an exponentially decaying bound. The proof of Theorem 1 is given in Appendix A, and the intuitive explanation of the $d_\infty$ assumption in the POMDP planning context is given in Section 4.2.2.

This exponential decay is important for the proofs in this section. We need to ensure that all nodes of the Sparse Sampling-$\omega$ tree at all depths $d$ reach convergence. The branching of the tree induces a factor proportional to $N^D$. A probabilistic bound that decays exponentially with $N$ will not only help offset the $N^D$ factor even with increasing depths, but also be consistent with Hoeffding-type bound exponential error rate that we also use to bound intermediate estimator errors.

4.2.2 Assumptions for Analyzing Sparse Sampling-$\omega$.

The following assumptions are needed for the Sparse Sampling-$\omega$ coupled convergence proof:

(i) $S$ and $O$ are continuous spaces, and the action space has a finite number of elements, $|A| < +\infty$.

(ii) For any observation sequence $\{o_n\}_{n=1}^d$, the densities $Z, T, b_0$ are chosen such that the Rényi divergence of the target distribution $\mathcal{P}^d$ and sampling distribution $\mathcal{Q}^d$ (Eqs. (30) and (31)) is bounded above by $d_\infty$ for all $d = 0, \ldots, D - 1$:

$$d_\infty(\mathcal{P}^d \| \mathcal{Q}^d) = \operatorname{ess} \sup_{x \sim \mathcal{Q}^d} W_{\mathcal{P}^d \| \mathcal{Q}^d}(x) \leq d_\infty^\text{max}$$  

(18)

(iii) The reward function $R$ is bounded by a finite constant $R_{\text{max}}$, and hence the value function is bounded by $V_{\text{max}} \equiv \frac{R_{\text{max}}}{1 - \gamma}$.

(iv) We can sample from the generating function $G$ and evaluate the observation probability density $Z$.

(v) The POMDP terminates after no more than $D < \infty$ steps.

Intuitively, condition (iii) means that the ratio of the conditional observation probability to the marginal observation probability cannot be too high. Additionally, the results still hold even when either of $S$ or $O$ are discrete, so long as it does not violate condition (ii) by appropriately switching the integrals to sums.

Although our analysis is restricted to the case when $\gamma < 1$ and the problem has a finite horizon, we believe that similar results can be derived for either when $\gamma = 1$ for a finite horizon or for infinite
horizon problems when $\gamma < 1$ by using the common argument that eventually future discounted rewards will be small (Kearns et al., 2002; Silver & Veness, 2010). Furthermore, while the results from this section repeat steps taken in proving POWSS (Lim et al., 2020), we significantly modify the details for Sparse Sampling-\(\alpha\).

### 4.2.3 Particle Likelihood Weighting Accuracy

As a precursor to Theorem 2, we establish a general result about function estimation using state particles with likelihood weights. This is useful because the inductive proof for showing Sparse Sampling-\(\alpha\) convergence in Lemma 2 relies heavily upon an SN estimator concentration inequality as well as a Hoeffding-type inequality.

**Lemma 1** (Particle Likelihood SN Estimator Convergence). Suppose a function \(f\) is bounded by a finite constant \(\|f\|_{\infty} \leq f_{\text{max}}\), and a particle belief state \(\tilde{b}_d = \{(s_{d,i}, w_{d,i})\}_{i=1}^{C}\) at depth \(d\) represents \(b_d\) with particle likelihood weighting that is recursively updated as \(w_{d,i} = w_{d-1,i} \cdot Z(o_d \mid a, s_d)\) for an arbitrary observation sequence \({o_n}\)\(_{n=1}^{d}\). Then, for all \(d = 0, \ldots, D - 1\), the following weighted average is the SN estimator of \(f\) under the belief \(b_d\) corresponding to observation sequence \({o_n}\)\(_{n=1}^{d}\):

\[
\hat{\mu}_{b_d}[f] = \frac{\sum_{i=1}^{C} w_{d,i} f(s_{d,i})}{\sum_{i=1}^{C} w_{d,i}},
\]

and the following concentration bound holds with probability at least \(1 - 3\exp(-C \cdot t_{\text{max}}^2(\lambda, C))\):

\[
|E_{s \sim b_d}[f(s)] - \hat{\mu}_{b_d}[f]| \leq \lambda,
\]

\[
t_{\text{max}}(\lambda, C) \equiv \frac{\lambda}{f_{\text{max}} d_{\text{max}}^2} - \frac{1}{\sqrt{C}}.
\]

**Proof.** We only outline the important steps here, and defer the detailed proof of this lemma to Appendix B. The key of this proof lies in the fact that the state particles trajectories \({s_{n,1}}\), \ldots, \({s_{n,C}}\) are i.i.d.r.v. sequences of depth \(d\), as GENPF independently generates each state sequence \(i\) according to the transition density \(T\). While GENPF generates highly correlated observation sequences and histories \({o_n}\)\(_{n=1}^{d}\), the dependence on observation sequence for a given particle belief state only comes through in the particle likelihood weights.

We abbreviate some terms of interest with the following notation:

\[
\mathcal{T}_{1:d}^i \equiv \prod_{n=1}^{d} T(s_{n,i} \mid s_{n-1,i}, a_n); \quad \mathcal{Z}_{1:d}^{i,j} \equiv \prod_{n=1}^{d} Z(o_{n,j} \mid a_n, s_{n,i}),
\]

where \(d\) is the depth, \(i\) is the index of the state sample, and \(j\) is the index of the observation sample. Absence of indices \(i, j\) means that \({s_n}\) and/or \({o_n}\) appear as regular variables. Intuitively, \(\mathcal{T}_{1:d}^i\) is the transition density of state sequence \(i\) from the root node to depth \(d\), and \(\mathcal{Z}_{1:d}^{i,j}\) is the conditional density of observation sequence \(j\) given state sequence \(i\) from the root node to depth \(d\). Additionally, \(b_d^i\) denotes \(b_d(s_{d,i})\) and \(w_{d,i}\) the weight of \(s_{d,i}\).

Then, we apply the importance sampling formalism to our system for all depths \(d = 0, \ldots, D - 1\). Here, \(\mathcal{P}^d\) is the normalized measure incorporating the probability of observation sequence \({o_n,j}\)\(_{n=1}^{d}\) on top of the state sequence \({s_{n,j}}\)\(_{n=1}^{d}\) until the node at depth \(d\), and \(\mathcal{Q}^d\) is the measure of the state sequence. We can think of \(\mathcal{P}^d\) corresponding to the observation sequence \({o_{n,j}}\). For simplicity,
we also denote $Z_{1:d}$ as the product of observation likelihoods $\prod_{n=1}^{d} Z(o_n \mid a_{n-1}, s_n)$ and $T_{1:d}$ as the product of transition densities $\prod_{n=1}^{d} T(s_n \mid s_{n-1}, a_{n-1})$. Then, for an arbitrary action sequence $\{a_n\}$, the following describes the densities necessary to define importance weighting:

$$
P^d = P^d_{\{a_n\}}(\{s_{n,i}\}) = \frac{(Z_{1:d}^{i,j})(T_{1:d}^{i})b_i^d}{\int S d0 d_{S0:d}}$$  \hspace{0.5cm} (23)

$$
Q^d = Q^d(\{s_{n,i}\}) = (T_{1:d}^{i})b_i^d$$  \hspace{0.5cm} (24)

$$
w_{P^d/Q^d}(\{s_{n,i}\}) = \frac{(Z_{1:d}^{i,j})}{\int S d0 d_{S0:d}}$$  \hspace{0.5cm} (25)

Here, the integral to calculate the normalizing constant is taken over $S^{d+1}$, the Cartesian product of the state space $S$ over $d+1$ steps. Now, we claim that the recursive likelihood updating scheme produces valid likelihood weights up to a normalization. With our recursive definition of the empirical weights, we obtain the full weight of each state sequence for a fixed observation sequence:

$$w_{d,i} = w_{d-1,i} \cdot Z(o_{d,j} \mid a_d, s_{d,i}) = \ldots = Z_{1:d}^{i,j} \propto w_{P^d/Q^d}(\{s_{n,i}\}).$$  \hspace{0.5cm} (26)

Consequently, we conclude that the weighted average with particle likelihood weights indeed corresponds to the proper SN estimator:

$$\bar{\mu}_{\pi_d}[f] = \frac{\sum_{i=1}^{C} w_{d,i} \cdot f(s_{d,i})}{\sum_{i=1}^{C} w_{d,i}} = \frac{\sum_{i=1}^{C} w_{P^d/Q^d}(\{s_{n,i}\}) \cdot f(s_{d,i})}{\sum_{i=1}^{C} w_{P^d/Q^d}(\{s_{n,i}\})}. \hspace{1cm} (27)$$

We can apply the SN concentration inequality in Theorem 1 to obtain the concentration bound. □

4.2.4 Coupled Convergence of Sparse Sampling-\(\omega\).

The theorem below describes Sparse Sampling-\(\omega\)’s coupled convergence to both optimal $Q$-values of the POMDP $Q^*_p$ and PB-MDP $Q^*_{\omega,p}$, as $C$ is increased.

**Theorem 2** (Sparse Sampling-\(\omega\) Coupled Optimality). Suppose conditions (I)-(V) are satisfied. Then, for any desired policy optimality $\varepsilon > 0$, choosing constants $C, \lambda, \delta$ that satisfy:

$$\lambda = \varepsilon(1-\gamma)^2/8,$$  \hspace{0.5cm} (28)

$$\delta = \lambda/(V_{\max} D (1-\gamma)^2),$$  \hspace{0.5cm} (29)

$$C = \max \left\{ \left( \frac{4 V_{\max} D_{\max}^2}{\lambda} \right)^2, \frac{64 V_{\max}^2}{\lambda^2} \left( D \log \frac{24 A |D_{\max}| V_{\max}^2 D}{\lambda^2} + \log \frac{1}{\delta} \right) \right\},$$  \hspace{0.5cm} (30)

the $Q$-function estimates $\hat{Q}^*_{\omega,d}(\bar{b}_d, a)$ obtained for all depths $d = 0, \ldots, D-1$ and all actions $a$ are jointly near-optimal with respect to $Q^*_p$ and $Q^*_{\omega,p}$ with probability at least $1 - \delta$:

$$|Q^*_p(b_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1-\gamma},$$  \hspace{0.5cm} (31)

$$|Q^*_{\omega,p}(\bar{b}_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1-\gamma}. \hspace{1cm} (32)$$
To prove Theorem 2 we follow a similar proof strategy from our previous proof for POWSS (Lim et al., 2020) to show that Eq. (31) holds, and a similar strategy of the original sparse sampling proof (Kearns et al., 2002) to show that Eq. (32) holds. In essence, this Sparse Sampling-ω convergence guarantee builds on POWSS and sparse sampling convergence guarantees, providing coupled convergence results to optimal $Q$-values of the POMDP $Q^*_P$ and PB-MDP $Q^*_M$.

First, we use induction in Lemma 2 to prove a concentration inequality for the value function at all nodes in the tree, starting at the leaves and proceeding up to the root. Consequently, proving Lemma 2 allows us to prove Theorem 2 with some justifications of how the parameter $C$ can actually be explicitly chosen with the choice of $\varepsilon$. The detailed proof for Theorem 2 is in Appendix D.

**Lemma 2 (Sparse Sampling-ω Estimator $Q$-Value Coupled Convergence).** For all $d = 0, \ldots, D - 1$ and $a$, the following bounds hold with probability at least $1 - 6|A|(4|A|C)^D \exp(-C \cdot \varepsilon^2)$:

\[
\begin{align*}
|Q^*_P(b_d, a) - \hat{Q}^*_P(b_d, a)| &\leq \alpha_d, \quad \alpha_d = \lambda + \gamma \alpha_{d+1}, \quad \alpha_{D-1} = \lambda, \quad (33) \\
|Q^*_M(b_d, a) - \hat{Q}^*_M(b_d, a)| &\leq \beta_d, \quad \beta_d = \gamma (\lambda + \beta_{d+1}), \quad \beta_{D-1} = 0, \quad (34) \\
t_{\max}(\lambda, C) &= \frac{\lambda}{4V_{\max}d_{\max}} - \frac{1}{\sqrt{C}}, \quad t = \min\{t_{\max}, \lambda / 4\sqrt{2V_{\max}}\} \quad (35)
\end{align*}
\]

**Proof.** We outline how we use the particle likelihood SN estimator inequality and Hoeffding inequality to bound the $Q$-values, and defer the detailed proof to Appendix C.

The optimal $d$-step $Q$-values for the POMDP $Q^*_P$ and the corresponding PB-MDP $Q^*_M$ are

\[
\begin{align*}
Q^*_P(b_d, a) &= \mathbb{E}_P[R(s_d, a) + \gamma V^*_P(b_{d+1}) | b_d] \\
&= \int \mathbb{E}_P[R(s_d, a)]d_{s_d} + \gamma \int \int V^*_P(b_{d+1})d_{s_d}d_{d+1}d_{o}, & (36) \\
Q^*_M(b_d, a) &= \mathbb{E}_P[V^*_M(b_{d+1}) | b_d, a] \\
&= \sum_{i=1}^C w_{d,i} \cdot R(s_{d,i}, a) + \gamma \int \sum_{i=1}^C V^*_M(b_{d+1}) \tau(b_{d+1} | b_d, a)dB_{d+1}. & (37)
\end{align*}
\]

The Sparse Sampling-ω value estimates are mathematically equal to

\[
\begin{align*}
\hat{V}^*_\omega_d(b_d) &= \max_{a \in A} \hat{Q}^*_\omega_d(b_d, a) \quad (40) \\
\hat{Q}^*_\omega_d(b_d, a) &= \sum_{i=1}^C w_{d,i} \hat{V}^*_\omega_{d,i} + \frac{1}{C} \sum_{i=1}^C \gamma \cdot \hat{V}^*_\omega_{d+1}(b_{d+1}^{|I_i|}), \quad (41)
\end{align*}
\]

where \{I_i\} are $C$ independent identically distributed random variables with finite discrete distribution $p_{w_{d}}$ with probability mass $p_{w_{d}}(I = i) = (w_{d,i} / \sum_k w_{d,k})$. Particle belief state $b_{d+1}^{|I_i|}$ is updated by an observation generated from $s_{d,i}$. This reflects the fact that GENPF randomly selects a state particle $s_o$ with probability $w_{d,o} / \sum_k w_{d,k}$ $C$ times independently to generate a new observation for the particle belief state after next step.

**POMDP Value Convergence:** First, we show that Eq. (33) is satisfied, which is an adapted and modified proof of POWSS convergence (Lim et al., 2020). Using the triangle inequality for a given step $d$ of the inductive proof, we split the difference into two terms, the reward estimation error (A)
and the next-step value estimation error (B):

\[
\left| Q_{\tilde{P},d}(b_d,a) - \tilde{Q}_{\tilde{P},d}(\tilde{b}_d,a) \right| \leq \left| \mathbb{E}_P[R(s_d,a) \mid b_d] - \frac{\sum_{d=1}^D \sum_{i=1}^C w_{d,i} r_{d,i}}{\sum_{d=1}^D \sum_{i=1}^C w_{d,i}} \right| + \gamma \left| \mathbb{E}_P[V_{\tilde{P},d+1}(ba) \mid b_d] - \frac{1}{C} \sum_{i=1}^C \bar{V}_{\tilde{P},d+1}(\tilde{b}_{d+1}) \right| \tag{42}
\]

The reward estimation error (A) is exactly the particle likelihood importance sampling error for estimating the reward function \( R(\cdot, a) \), which can be bounded by applying Lemma\[1\]

To bound the next-step value estimation error (B), we introduce a particle likelihood SN estimator to bridge the following quantities (detailed definitions and bounds of each term are in Appendix [D]):

\[
\left| \mathbb{E}_P[V_{\tilde{P},d+1}(ba) \mid b_d] - \frac{1}{C} \sum_{i=1}^C \bar{V}_{\tilde{P},d+1}(\tilde{b}_{d+1}) \right| \leq \left| \mathbb{E}_P[V_{\tilde{P},d+1}(ba) \mid b_d] - \frac{\sum_{i=1}^C w_{d,i} V_{\tilde{P},d+1}(b_d,a)^i}{\sum_{d=1}^D \sum_{i=1}^C w_{d,i}} \right| + \frac{1}{C} \sum_{i=1}^C \left| \bar{V}_{\tilde{P},d+1}(\tilde{b}_{d+1}) - \frac{1}{C} \sum_{i=1}^C V_{\tilde{P},d+1}(b_d,a)^i \right| \tag{43}
\]

(1) Importance sampling error

(2) MC weighted sum approximation error

(3) MC next-step integral approximation error

(4) Function estimation error

**PB-MDP Value Convergence:** Second, we show that Eq. (34) is satisfied, which is an adapted and modified proof of sparse sampling convergence (Kearns et al., 2002). Once again, we split the difference between the SN estimator and the \( Q_{\tilde{P},d}^* \) function into two terms, the reward estimation error (A) and the next-step value estimation error (B):

\[
\left| Q_{\tilde{P},d}^*(\tilde{b}_d,a) - \tilde{Q}_{\tilde{P},d}^*(\tilde{b}_d,a) \right| \leq \left| \rho(\tilde{b}_d,a) - \rho(\tilde{b}_d,a) \right| \tag{44}
\]

(43)

(44)

Since our particle belief MDP induces no reward estimation error, the term (A) is always 0 and proving the base case \( d = D - 1 \) is trivial as (A) and (B) are both 0. Then, we show that the difference (B) is bounded for all \( d = 0, \ldots, D - 1 \). We use the triangle inequality repeatedly to separate it into two terms; (1) the MC transition approximation error, and (2) the function approximation error
O(\exp(-t \cdot C)) probability factor, where \( t \) is some constant, as both the SN concentration bound and the Hoeffding bound are exponentially decaying. Since this upper bound on the estimation error needs to hold for all steps \( d = 0, \ldots, D-1 \), we must apply the worst case union bound on the probability to ensure that every node in the tree achieves the desired concentration bound. This results in a worst case probability factor that is \( O(C^D) \). Therefore, we can obtain the \( Q \)-value estimator concentration inequality, with convergence rate \( O(C^D \exp(-t \cdot C)) \).

5. Particle Belief MDP Approximation Guarantees

In this section, we establish the theoretical guarantees for using any approximately optimal MDP planning algorithm to solve the POMDP problem \( P \) by planning in the particle belief MDP \( M_p \). Theorem 3 shows that the \( Q \)-values \( Q_P^* \) and \( Q_{M_p}^* \) are close to each other with high probability, and Theorem 4 shows that using any approximately optimal MDP planning algorithm \( A \) in the particle belief MDP \( M_p \) as a policy yields near-optimal value in the original POMDP with an additional closed-loop exact belief update step.

5.1 Particle Belief MDP \( Q \)-Value Approximation Optimality

We introduce Theorem 3 which probabilistically bridges the POMDP \( P \) and its corresponding particle belief MDP \( M_p \). In essence, this theorem claims that the two optimal \( Q \)-values \( Q_P^* \) and \( Q_{M_p}^* \) are close with high probability, due to the virtue of creating a very accurate value estimation via Sparse Sampling-\( \omega \) that is close to both \( Q_P^* \) and \( Q_{M_p}^* \), happens with high probability.

**Theorem 3** (Particle Belief MDP \( Q \)-Value Approximation Optimality). Given a finite horizon POMDP \( P \) and its corresponding particle belief MDP \( M_p \), there exists a number of particles \( C \) for which the optimal \( Q \) value of the POMDP problem \( Q_P^*(b, a) \) can be approximated by the optimal \( Q \) value of the particle belief MDP problem \( Q_{M_p}^*(\bar{b}, a) \) with arbitrary precision. Namely, under the regularity conditions \( \{i\} \{iv\} \) the following bound holds for a given belief \( b \), corresponding sampled particle belief \( \bar{b} \), and all available actions \( a \) with probability at least \( 1 - \delta_{M_p} \) for a desired accuracy \( \varepsilon_{M_p} \):

\[
|Q_P^*(b, a) - Q_{M_p}^*(\bar{b}, a)| \leq \varepsilon_{M_p}.
\]

**Proof.** The main idea of the proof is that we bridge the two \( Q \)-values, \( Q_P^* \) and \( Q_{M_p}^* \), via approximation through Sparse Sampling-\( \omega \) with \( C \) particles. From Theorem 2 we have established that there exists an algorithm, Sparse Sampling-\( \omega \), which is jointly optimal in both senses of POMDP \( P \).
and its corresponding particle belief MDP $M_p$. Then, if we were to hypothetically perform Sparse Sampling-$\omega$ of depth $D$, the sum of the errors between the three types of $Q$-values at the root node, $Q_P^*$, $Q_{M_p}$, and $\hat{Q}_{\omega}$, are jointly bounded with probability at least $1 - \tilde{\delta}_{M_p}$ through Theorem 2 where $\tilde{\delta}_{M_p}$ follows the same definition as $\delta$. We use the fact that $Q_P^*$ and $Q_{M_p}$ are the optimal $Q$-values at $d = 0$ for the POMDP and PB-MDP, respectively:

$$|Q_P^*(b,a) - Q_{M_p}^*(\bar{b},a)| \leq |Q_P^*(b,a) - \hat{Q}_{\omega}^*(\bar{b},a)| + |\hat{Q}_{\omega}^*(\bar{b},a) - Q_{M_p}^*(\bar{b},a)|$$

$$\equiv |Q_{P,0}^*(b_0,a) - \hat{Q}_{\omega,0}^*(b_0,a)| + |Q_{M_p,0}^*(b_0,a) - \hat{Q}_{\omega,0}^*(\bar{b}_0,a)|$$

$$\leq \frac{2\lambda}{1-\gamma} (\equiv \varepsilon_{M_p}).$$

Here, instead of using the $\lambda$ definition in Theorem 2 we can directly choose $\lambda$ such that the above quantity is equal to $\varepsilon_{M_p}$, since $\lambda$ only depends on a desired approximation error $\varepsilon$ and $\gamma$. However, we emphasize the intermediate step of the $Q$-value difference bounded by $\frac{2\lambda}{1-\gamma}$ to invoke this fact later on in the policy convergence proof of Theorem 3. Since this bound holds with high probability for creating any hypothetical Sparse Sampling-$\omega$ tree, this must mean that $|Q_P^*(b,a) - Q_{M_p}^*(\bar{b},a)| \leq \varepsilon_{M_p}$ in general with high probability.

The convergence rate $\tilde{\delta}_{M_p}$ is $O(C^D \exp(-\gamma \cdot C))$. This means that as we increase the number of particles, we can expect better performance by approximately solving a POMDP via particle belief MDP planning approximation.

5.2 Particle Belief MDP Planning Optimality

**Corollary 1** (Particle Belief MDP Planning Optimality). Under regularity conditions necessary for both the particle belief MDP and an MDP planning algorithm $A$, if the optimal planner can approximate $Q$ values with arbitrary precision $\varepsilon_A$ with probability at least $1 - \delta_A$ in the corresponding particle belief MDP of a given POMDP, then the planning algorithm can approximate the POMDP $Q$-values within $\varepsilon_{M_p} + \varepsilon_A$ with probability at least $1 - \delta_{M_p} - \delta_A$:

$$|Q_P^*(b,a) - \hat{Q}_{M_p}^*(\bar{b},a)| \leq \varepsilon_{M_p} + \varepsilon_A.$$

**Proof.** This is a straightforward application of triangle inequality for the $Q$-value estimation accuracy and worst case union bound for the probability.

Essentially, this corollary means that we can use any approximately optimal MDP planning algorithm to solve the POMDP problem by planning in the particle belief MDP instead, and still retain similar optimality guarantees. In most practical cases, this method would usually incur an additional $O(C)$ compute time factor as single particle belief state generation now needs to propagate $C$ particles forward instead of a single particle/state. Note that we can also devise an expected value version of the bounds by converting the probability statement into an expected value statement.

Consequently, proving Corollary 1 allows us to prove Theorem 3 with additional results from Kearns et al. (2002) and Singh and Yee (1994). Through the near-optimality of the $Q$-functions, we conclude that the value obtained by employing a near-optimal MDP policy in the PB-MDP is also near-optimal in the original POMDP with further assumptions on the closed-loop POMDP system. The detailed proof for Theorem 3 is in Appendix E.
Theorem 4 (Particle Belief MDP Approximate Policy Convergence). In addition to regularity conditions for particle belief MDP and the MDP planning algorithm $A$, assume that the closed-loop POMDP Bayesian belief update step is exact. Then, for any $\varepsilon > 0$, we can choose a $C$ such that the value obtained by planning with $A$ in the PB-MDP is within $\varepsilon$ of the optimal POMDP value function at $b_0$:

$$V^*_P(b_0) - V^*_M(b_0) \leq \varepsilon. \quad (51)$$

5.3 Sparse Particle Filter Tree (Sparse-PFT)

By utilizing the results in Theorem 3 and Theorem 4, we can promote a sampling-based MDP planning algorithm Upper Confidence Tree (UCT) into Sparse Particle Filter Tree (Sparse-PFT) and retain similar convergence guarantees for the POMDP (Bjarnason et al., 2009; Kocsis & Szepesvári, 2006; Shah et al., 2022). This results in a simple algorithm that is simple to implement, and enjoys both theoretical guarantees and practicality.

The core of Sparse-PFT is the SIMULATE function defined in Algorithm 3 that is called repeatedly to construct the tree. The set of global variables for Sparse-PFT includes the same global variables used for Sparse Sampling-$\omega$ with the addition of $n$, the number of tree search queries, and $c_{UCB}$, the Upper Confidence Bound critical factor that determines the amount of exploration. The SIMULATE function is analogous to the function of the same name from UCT (Bjarnason et al., 2009; Kocsis & Szepesvári, 2006), with the only difference being Sparse-PFT manages particle belief sets through GENPF rather than states directly.

In the above algorithm definition, $C(\cdot)$ represents the list of children nodes, $N(\cdot)$ the number of visits to the node, $Q(\cdot)$ the estimated $Q$-value at the node, and $c_{UCB}$ the Upper Confidence Bound exploration parameter. These lists are all implicitly initialized to 0 or $\emptyset$. The ROLLOUT procedure is an optional heuristic that runs a simulation with a heuristic rollout policy for $d$ steps to estimate the value, while avoiding building a large computation tree at each step of simulation.

With the introduction of Sparse-PFT, we can view the recent POMDP algorithms as practical extensions of Sparse-PFT. For instance, PFT-DPW (Sunberg & Kochenderfer, 2018) is a simple
Algorithm 3 Sparse-PFT Algorithm

Global Variables: $\gamma, n, c_{UB}, G, D$.

Algorithm: SIMULATE($\bar{b}, d$)

Input: particle belief set $\bar{b} = \{(s_i, w_i)\}$, depth $d$.

Output: A scalar $q$ that is the total discounted reward of one simulated trajectory sample.

1: if $d = D$ then
2: return $0$
3: $a \leftarrow \arg\max_{a \in C(\bar{b})} Q(\bar{b}, a) + c_{UB} \sqrt{\frac{\log N(\bar{b})}{N(\bar{b}, a)}}$
4: if $|C(\bar{b}, a)| = C$ then
5: $\bar{b}', \rho \leftarrow$ sample from $C(\bar{b}, a)$
6: else
7: $\bar{b}', \rho \leftarrow$ GENPF($\bar{b}, a$)
8: $C(\bar{b}, a) \leftarrow C(\bar{b}, a) \cup \{(\bar{b}', \rho)\}$
9: if $N(\bar{b}) = 0$ then
10: $q \leftarrow \rho + \gamma \cdot$ ROLLOUT($\bar{b}', d - 1$)
11: else
12: $q \leftarrow \rho + \gamma \cdot$ SIMULATE($\bar{b}', d - 1$)
13: $N(\bar{b}) \leftarrow N(\bar{b}) + 1$
14: $N(\bar{b}, a) \leftarrow N(\bar{b}, a) + 1$
15: $Q(\bar{b}, a) \leftarrow Q(\bar{b}, a) + \frac{q - Q(\bar{b}, a)}{N(\bar{b}, a)}$
16: return $q$

A modification of Sparse-PFT by utilizing the double progressive widening (DPW) technique to additionally handle continuous action spaces, and POMCPow (Sunberg & Kochenderfer, 2018) is a further extension that plans based on particle trajectory that allows for flexible particle number representations of a given belief node. However, further theoretical analyses of these algorithms would most likely require more sophisticated techniques and further assumptions.

6. Numerical Experiments

Numerical simulation experiments were conducted in order to evaluate and compare the performances of our new simple algorithm, Sparse-PFT, along with other solvers. In particular, we also ran experiments for Adaptive online packing-guided search (AdaOPS) (Wu et al., 2021), a recent solver with practical performance and partial theoretical guarantees. We also show performances of other hallmark algorithms like QMDP and POMCP along with random policy to demonstrate the need for continuous observation POMDP solvers that can handle more general assumptions.

In all five of the numerical experiments shown in Fig. 3, the POMDP solvers were limited to at most 1 second of planning time per step. For the closed loop planning, the belief updates were accomplished with a particle filter independent of the planner, and no part of the planning tree was saved for re-use on subsequent steps. A total of 5000 simulation experiments were conducted for each configuration combination of solver and environment in order to obtain the Monte Carlo mean and standard error estimates for the laser tag and VDP tag environments, and 1000 simulation experiments for the light dark and Sub Hunt problems since planners typically yielded more consistent performances for these problems. The tabular summary of all results is given in Table 1 and corresponding figure summary of all results for different planning time allotments is given in Fig. 4.
Table 1: Tabular comparative benchmark summary. All results are given as (mean ± standard error) for a 1 second planning time allotment. The algorithm with the best average performance is shown in boldface for each experiment. The three letters after each algorithm name indicate whether the state, action, and observation spaces are continuous or discrete.

The open source code for the experiments is built on the POMDPs.jl framework (Egorov et al., 2017), and is available at: github.com/WhiffleFish/PFTExperiments. The hyperparameter values used for the experiments are shown in Appendix F. The following sections contain descriptions of the evaluation problems along with discussion of solver performances.

6.1 Laser Tag

The laser tag POMDP is taken from the DESPOT benchmarks (Ye et al., 2017) wherein a robot is required to use laser sensors to localize with the ultimate goal of catching an evading robot. The agent’s laser sensors extend radially in 8 evenly spaced directions and each return a rounded sensed distance sampled from a normal distribution given by $N(d, 2.5)$ where $d$ is the true distance to the nearest obstacle. Although the observation space is not continuous, it is sufficiently large (on the order of $10^6$) that most online solvers would have to treat this as close to continuous.

From the results, we find that POMCPOW outperforms the PFT methods for lower planning times due to a significantly lower node expansion cost. While POMCPOW only needs to
We note that POMCP particularly struggles on this problem compared to all other algorithms. The large observation space forces the trees constructed by POMCP to become extremely shallow due to each unique sampled observation resulting in a new leaf node. This hinders POMCP’s ability to develop a non-myopic multi-step plan and yield accurate action values, empirically showing the importance of particle weighting. On the other hand, QMDP performs similarly well as the modern solvers. While the agent does not initially know its own location, it has sufficient enough information to localize using the laser sensor observations after some number of steps, and the evading robot behavior is not stochastic. Therefore, the crude QMDP approximation sufficiently performs well, seemingly due to the problem requiring less active information gathering compared to the other problems.
6.2 Light Dark

The 1-dimensional light dark POMDP revolves around the requirement of active information gathering. The state is an integer representing the position of the agent and the action space is \( \mathcal{A} = \{-10, -1, 0, 1, 10\} \). Deterministic transitions are given by \( s' = s + a \). The reward,

\[
R(s, a) = \begin{cases} 
+100 & \text{if } s = 0, a = 0 \\
-100 & \text{if } s \neq 0, a = 0 \\
-1 & \text{otherwise}
\end{cases}
\]  

(52)

dictates that the optimal policy drive the state to the origin as quickly as possible. Because the state is not immediately known, inferences over the true state must be made over noisy observations that grow in variance proportional to the agent’s distance from the light location at...
The observation distribution is $\mathcal{N}(s, |s - 10| + \varepsilon)$, where $\varepsilon$ is some small constant included to prevent observation weights from reaching $+\infty$ due to a collapse to a Dirac distribution when the agent arrives at the light location.

The planners that yield the highest expected reward in the light dark domain roughly follow a 2-step plan: first localizing at the light location, then traveling down to the goal location. Essentially, the light location becomes a necessary subgoal. We can demonstrate this by creating a heuristic policy that initially steers towards the light location via certainty-equivalent control, and then takes action $a = -10$ down to the true goal location. This heuristic policy yields an expected reward of $62.0 \pm 0.19$ which is as good or better than any planner shown in Table I.

Surprisingly, higher planning times do not necessarily correspond to increasing expected rewards in the light dark domain. For AdaOPS, the solver converges to its peak expected reward with a planning time as low as 0.01 seconds leading to marginal improvement with further increases in planning time. Within a planning time interval of $[0.03, 0.1]$ seconds, the performance of POMCPOW decreases, indicating that the planner becomes increasingly confident in a suboptimal plan. One possible source of this overconfidence is beliefs represented by a single particle. This same behavior becomes evident in PFT planners with single rollout value estimation. However, by increasing the number of sampled particles that are chosen as the true state in belief-based rollouts, the belief value estimate is granted lower variance and greater accuracy, effectively reducing the time spent exploring suboptimal branches of the constructed tree.

Because of this necessity to decrease belief entropy before committing to the goal, QMDP performs suboptimally due to no emphasis being placed on costly information gathering. Therefore, regardless of belief distribution entropy, QMDP myopically steers directly towards the goal location but rarely commits within the simulation horizon due to high state uncertainty.

### 6.3 Sub Hunt

In the Sub Hunt POMDP, from the POMCPOW benchmark (Sunberg & Kochenderfer, 2018), the agent controls a submarine with the goal of finding and destroying an opposing submarine. The state space consists of the grid locations of both the agent and enemy submarines, a Boolean determining whether or not the enemy is aware of the agent’s presence, and the enemy’s goal direction ($\{1, \ldots, 20\}^4 \times \{\text{aware, unaware}\} \times \{N, S, E, W\}$). The agent is given the option to move three steps in any of the four cardinal directions, attack the enemy, or ping the enemy with active sonar while the enemy randomly chooses between taking two steps forward or one step diagonally forward.

In the Sub Hunt domain, PFT methods dominate all other planners over all planning times. Because the state space is discrete, value iteration can be used to calculate $Q$-values for the fully observable MDP, and QMDP (Littman et al., 1995) can be used for the rollout policy. With this strong belief-based rollout policy, both PFT-
DPW and Sparse-PFT are able to construct nearly-optimal policies with planning times as low as 0.01 seconds, leading to no noticeable further improvement over longer planning times. Conversely, POMCPOW and AdaOPS have gradually increasing planning curves in Fig. 4 with AdaOPS nearly reaching the performance of PFT planners at 1 second and POMCPOW reaching an earlier inflection point, resulting in a final performance lower than the other three planners.

6.4 VDP Tag

The Van Der Pol Tag POMDP formulation tasks the agent with moving through a two-dimensional space to catch an opponent whose dynamics are governed by the Van Der Pol differential equations,

\[
\dot{x} = \mu \left( x - \frac{x^3}{3} - y \right), \quad \dot{y} = \frac{x}{\mu},
\]

for which we use scaling constant \( \mu = 2 \). Because this problem has a continuous state space (\( S = \mathbb{R}^4 \)), a continuous action space (\( A = [0,2\pi] \times \{0,1\} \)) and a continuous observation space (\( O = \mathbb{R}^8 \)), discrete value iteration is no longer admissible as input for a value estimation policy. AdaOPS is unable to handle continuous action spaces thus it is omitted from this benchmark. The Van Der Pol tag domain is the first in which there exists a noticeable performance gap between Sparse-PFT and PFT-DPW, indicating that action progressive widening offers some utility over fixed widening in continuous action space problems.

For the action-discretized Van Der Pol Tag, the available movement directions are 20 evenly spaced angles from 0deg to 360deg. For this discrete problem, we come across two new peculiarities: AdaOPS performance decreases with increased planning time, and PFT methods perform orders of magnitude worse than other planners at very low allotted planning times. This is likely attributable to a large action space (\( |A| = 20 \)) and a relatively expensive simulation function (RK4 integration). Because PFT methods propagate a collection of particles upon tree expansion, belief value estimates tend to be more accurate at the cost of added computation scaling linearly with the number of particles representing each belief. Thus, expanding all possible actions while using an computationally expensive simulator on all particles takes a long time, leading to very shallow trees.

7. Conclusion

In this work, we formally justify that optimality guarantees in a finite sample particle belief MDP (PB-MDP) approximation of a POMDP/belief MDP yields optimality guarantees in the original POMDP as well, which allows for simple yet powerful adaptations of MDP algorithms to solve POMDPs. By proving that the Sparse Sampling-\( \omega \) \( Q \)-value estimates are close to both optimal
\(Q\)-values of the POMDP and PB-MDP with high probability, we conclude that the optimal \(Q\)-values of the POMDP and PB-MDP themselves are close with high probability. Such fundamental bridge between PB-MDPs and POMDPs allows us to adapt any sampling-based MDP algorithm of choice to a POMDP by solving the corresponding particle belief MDP approximation and preserve the convergence guarantees in the POMDP, which only increases the computational complexity by a factor of \(O(C)\), by using particle filtering-based generative models. This motivates usage of particle belief-based MDP algorithms such as Sparse Particle Filter Tree (Sparse-PFT), which enjoys algorithmic simplicity, theoretical guarantees and practicality, as it is a variant of upper confidence trees (UCT) (Bjarnason et al., 2009; Shah et al., 2022) for PB-MDPs.

There are many interesting avenues for future research. First, the broader theoretical justification of more complex algorithms, such as PFT-DPW and POMCPOW, still do not exist. Showing theoretical validity of these algorithms would help close the gap between theory and practice even further. In addition, as seen in our numerical experiments, the best performing algorithm varies across different types of benchmarks. Further theoretical and empirical characterization of which algorithms are most effective for which problems could greatly aid practitioners. Lastly, while the algorithms presented here perform well in low dimensional continuous observation spaces, tree search for more difficult POMDPs, such as those with high dimensional observations (Deglurkar et al., 2021) and continuous/hybrid action spaces (Lim et al., 2021; Mern et al., 2021; Seiler et al., 2015) is more difficult, and further analytical and empirical research is warranted.

**Acknowledgments**

This material is based upon work supported by a DARPA Assured Autonomy Grant, the SRC CONIX program, NSF CPS Frontiers, the University of Colorado Boulder, and the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1752814. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any sponsoring organizations.
Appendix A. Proof of Theorem 1 - SN $d_\infty$-Concentration Bound

**Theorem 1** (SN $d_\infty$-Concentration Bound). Let $\mathcal{P}$ and $\mathcal{Q}$ be two probability measures on the measurable space $(\mathcal{X}, \mathcal{F})$ with $\mathcal{P}$ absolutely continuous w.r.t. $\mathcal{Q}$ and $d_\infty(\mathcal{P}||\mathcal{Q}) < +\infty$. Let $x_1, \ldots, x_N$ be independent identically distributed random variables (i.i.d.r.v.) with distribution $\mathcal{P}$, and $f : \mathcal{X} \to \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $\lambda > 0$ and $N$ large enough such that $\lambda > \|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q})/\sqrt{N}$, the following bound holds with probability at least $1 - 3\exp(-N \cdot t^2(\lambda, N))$:

$$\|E_{x \sim \mathcal{P}}[f(x)] - \hat{\mu}_{\mathcal{P}/\mathcal{Q}]| \leq \lambda, (1)$$

$$t(\lambda, N) = \|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q}) - \frac{1}{\sqrt{N}}. (2)$$

**Proof.** This proof follows similar proof steps as in Metelli et al. (Metelli et al., 2018). Since we have upper bounds on the infinity Rényi divergence $d_\infty(\mathcal{P}||\mathcal{Q})$, we can start from Hoeffding’s inequality for bounded random variables applied to the regular IS estimator $\hat{\mu}_{\mathcal{P}/\mathcal{Q}] = \frac{1}{N} \sum_{i=1}^{N} w_{\mathcal{P}/\mathcal{Q}}(x_i) f(x_i)$, which is unbiased. While applying Hoeffding’s inequality, we can view importance sampling on $f(x)$ weighted by $w_{\mathcal{P}/\mathcal{Q}}(x)$ as Monte Carlo sampling on $g(x) = w_{\mathcal{P}/\mathcal{Q}}(x) f(x)$, which is a function bounded by $\|g\|_\infty = d_\infty(\mathcal{P}||\mathcal{Q}) \|f\|_\infty$:

$$\mathbb{P}\left(\hat{\mu}_{\mathcal{P}/\mathcal{Q}] - E_{x \sim \mathcal{P}}[f(x)] \geq \lambda \right) \leq \mathbb{P}\left(\hat{\mu}_{\mathcal{P}/\mathcal{Q}] - E_{x \sim \mathcal{P}}[\hat{\mu}_{\mathcal{P}/\mathcal{Q}] f(x)] \geq \lambda \right) \leq \exp\left(-\frac{2N^2 \lambda^2}{\sum_{i=1}^{N} 2(d_\infty(\mathcal{P}||\mathcal{Q}) \|f\|_\infty)^2} \right) \leq \exp\left(-\frac{N \lambda^2}{d_\infty(\mathcal{P}||\mathcal{Q}) \|f\|_\infty^2} \right) \equiv \delta \leq 2\exp\left(-\frac{N \lambda^2}{d_\infty(\mathcal{P}||\mathcal{Q}) \|f\|_\infty^2} \right) = 2\delta.

(3) (4) (5) (6)

We prove a similar bound for the SN estimator $\tilde{\mu}_{\mathcal{P}/\mathcal{Q}] = \sum_{i=1}^{N} \tilde{w}_{\mathcal{P}/\mathcal{Q}}(x_i) f(x_i)$, which is a biased estimator. However, we need to take a step further and analyze the absolute difference, requiring us to split the difference up into two terms:

$$\mathbb{P}(\|E_{x \sim \mathcal{P}}[f(x)] - \tilde{\mu}_{\mathcal{P}/\mathcal{Q}]| \geq \lambda \right) \leq \mathbb{P}(\tilde{\mu}_{\mathcal{P}/\mathcal{Q}] - E_{x \sim \mathcal{P}}[f(x)] \geq \lambda \right) + \mathbb{P}(E_{x \sim \mathcal{P}}[f(x)] - \tilde{\mu}_{\mathcal{P}/\mathcal{Q}] \geq \lambda \right) \leq \mathbb{P}(\tilde{\mu}_{\mathcal{P}/\mathcal{Q}] - E_{x \sim \mathcal{P}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}] | \geq \lambda \right) + \mathbb{P}(E_{x \sim \mathcal{P}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}] - \tilde{\mu}_{\mathcal{P}/\mathcal{Q}] \geq \lambda \right) \leq \tilde{\delta} + \mathbb{P}(E_{x \sim \mathcal{P}}[f(x)] - \tilde{\mu}_{\mathcal{P}/\mathcal{Q}] \geq \lambda \right)

(7) (8) (9) (10)

The first term is bounded by $\tilde{\delta}$ from the above bound and recasting $\lambda$ to $\tilde{\lambda}$ to account for the bias of the SN estimator:

$$\tilde{\lambda} = \lambda - \|E_{x \sim \mathcal{P}}[f(x)] - E_{x \sim \mathcal{P}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}]| \right) \leq \lambda - \|E_{x \sim \mathcal{P}}[f(x)] - E_{x \sim \mathcal{P}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}]| \right) \left(11\right)$$

$$\tilde{\delta} = \exp\left(-\frac{N \tilde{\lambda}^2}{\tilde{\delta}^2(\mathcal{P}||\mathcal{Q}) \|f\|_\infty^2} \right)$$

(12)
Here, we define $t(\lambda, N)$ later on such that the $1/\sqrt{N}$ factor is nicely separated. We use the assumption in the theorem that $N$ is chosen large enough that $\lambda > \|f\|_\infty d_\infty(P || Q) / \sqrt{N}$ to bound the $\hat{\delta}$ term:

$$\hat{\delta} \leq \exp\left(-\frac{N(\lambda - \|f\|_\infty d_\infty(P || Q) / \sqrt{N})^2}{d_\infty^2(P || Q) \|f\|_\infty^2}\right)$$

$$= \exp\left(-N\left(\frac{\lambda - \|f\|_\infty d_\infty(P || Q) / \sqrt{N}}{\|f\|_\infty d_\infty(P || Q)}\right)^2\right)$$

$$\equiv \exp\left(-N \cdot t^2(\lambda, N)\right)$$

Here, we define $t(\lambda, N) \equiv \frac{\lambda}{\|f\|_\infty d_\infty(P || Q)} - \frac{1}{\sqrt{N}}$, which satisfies $0 < t(\lambda, N) \leq \frac{\lambda}{\|f\|_\infty d_\infty(P || Q)}$. The second term can be bounded similarly by rebounding the bias term with $\hat{\lambda}$, using symmetry and Hoeffding’s inequality:

$$P(\mathbb{E}_{x \sim P}[f(x)] - \bar{\mu}_{P/Q} \geq \hat{\lambda}) \leq P(\mathbb{E}_{x \sim Q}[\bar{\mu}_{P/Q}] - \bar{\mu}_{P/Q} \geq \hat{\lambda})$$

$$\leq P(|\mathbb{E}_{x \sim Q}[\bar{\mu}_{P/Q}] - \bar{\mu}_{P/Q}| \geq \hat{\lambda}) \leq 2\delta$$

Thus, we obtain the following bound:

$$P(0)$$
Appendix B. Proof of Lemma 1 (Continued) - Particle Likelihood SN Estimator Convergence

In the main paper, we show that \( \mu_{b_d}[f] \) is an SN estimator of \( \mathbb{E}_{s \sim b_d}[f(s)] \). We apply the concentration inequality proven in Theorem 1 to finish the proof of Lemma 1.

**Lemma 1** (Particle Likelihood SN Estimator Convergence). Suppose a function \( f \) is bounded by a finite constant \( ||f||_\infty \leq f_{\text{max}} \), and a particle belief state \( b_d = \{s_{d,i}, w_{d,i}\} \) at depth \( d \) that represents \( b_d \) with particle likelihood weighting that is recursively updated as \( w_{d,i} = w_{d-1,i} \cdot Z(o_d | a, s_{d-1}) \) for an observation sequence \( \{o_n\}_{n=1}^d \). Then, for all \( d = 0, \ldots, D-1 \), the following weighted average is the SN estimator of \( f \) under the belief \( b_d \) corresponding to observation sequence \( \{o_n\}_{n=1}^d \):

\[
\mu_{b_d}[f] = \frac{\sum_{i=1}^C w_{d,i} f(s_{d,i})}{\sum_{i=1}^C w_{d,i}},
\]

and the following concentration bound holds with probability at least \( 1 - 3 \exp(-C \cdot t_{\text{max}}^2(\lambda, C)) \),

\[
||\mathbb{E}_{s \sim b_d}[f(s)] - \mu_{b_d}[f]|| \leq \lambda
\]

\[
t_{\text{max}}(\lambda, C) = \frac{\lambda}{f_{\text{max}} d_{\text{max}}^{C}} - \frac{1}{\sqrt{C}}
\]

**Proof.** In this proof, we will take advantage of the fact that the state particles trajectories \( \{s_n\}_1, \ldots, \{s_n\}_C \) of depth \( d \) are independent of each other, as GENPF independently generates each state sequence \( i \) according to the transition density \( T \).

In the subsequent analysis, we abbreviate some terms of interest with the following notation:

\[
\mathcal{T}_{1:d}^i \equiv \prod_{n=1}^d \mathcal{T}(s_{n,i} | s_{n-1,i}, a_n); \quad \mathcal{Z}_{1:d}^{i,j} \equiv \prod_{n=1}^d \mathcal{Z}(o_{n,j} | a_n, s_{n,i}).
\]

Here \( d \) denotes the depth, \( i \) denotes the index of the state sample, and \( j \) denotes the index of the observation sample. Absence of indices \( i, j \) means that \( \{s_n\} \) and/or \( \{o_n\} \) appear as regular variables. Intuitively, \( \mathcal{T}_{1:d}^i \) is the transition density of state sequence \( i \) from the root node to depth \( d \), and \( \mathcal{Z}_{1:d}^{i,j} \) is the conditional density of observation sequence \( j \) given state sequence \( i \) from the root node to depth \( d \). Additionally, \( b^i_d \) denotes \( b_d(s_{d,i}) \) and \( w_{d,i} \) the weight of \( s_{d,i} \).

First, we show that \( \mu_{b_d}[f] \) is an SN estimator of \( \mathbb{E}_{s \sim b_d}[f(s)] \). By following the recursive belief update, the belief term can be fully expanded:

\[
b_{D-1}(s_{D-1}) = \frac{\int_{s_{D-1}} (\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-2}}{\int_{s_{D-1}} (\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-1}}
\]

Then, \( \mathbb{E}_{s \sim b_d}[f(s)] \) is equal to the following:

\[
\mathbb{E}_{s \sim b_d}[f(s)] = \int_{s} f(s_{D-1}) b_{D-1} ds_{D-1} = \frac{\int_{s_{D-1}} f(s_{D-1})(\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-1}}{\int_{s_{D-1}} (\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-1}}
\]

We approximate the \( \mathbb{E}_{s \sim b_d}[f(s)] \) function with importance sampling by using problem requirement (iv) where the target density is \( b_{D-1} \). First, we sample the sequences \( \{s_{n,i}\} \) according to the joint probability \( (\mathcal{T}_{1:D-1}) b_0 \). Afterwards, we weight the sequences by the corresponding observation density \( \mathcal{Z}_{1:D-1} \), obtained from the generated observation sequences \( \{o_{n,j}\} \). Normally, these generated
observation sequences through GENPF will be correlated. For now, we assume the observation sequences \( \{o_{n,j}\} \) are fixed.

Applying the importance sampling formalism to our system for all depths \( d = 0, \ldots, D - 1 \), \( \mathcal{P}^d \) is the normalized measure incorporating the probability of observation sequence \( j \) on top of the state sequence \( i \) until the node at depth \( d \), and \( \mathcal{Q}^d \) is the measure of the state sequence. We can think of \( \mathcal{P}^d \) corresponding to the observation sequence \( \{o_{n,j}\} \).

\[
\mathcal{P}^d = \mathcal{P}^d_{\{o_{n,j}\}}(\{s_{n,i}\}) = \frac{(Z_{1:d}^i)(T_{1:d}^i)b_0^i}{\int_{S^{d+1}}(Z_{1:d}^i)(T_{1:d}^i)b_0dS_{0,d}} \tag{30}
\]

\[
\mathcal{Q}^d = \mathcal{Q}^d(\{s_{n,i}\}) = (T_{1:d}^i)b_0^i \tag{31}
\]

\[
w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}) = \frac{(Z_{1:d}^i)}{\int_{S^{d+1}}(Z_{1:d}^i)(T_{1:d}^i)b_0dS_{0,d}} \tag{32}
\]

Here, the integral to calculate the normalizing constant is taken over \( S^{d+1} \), the Cartesian product of the state space \( S \) over \( d + 1 \) steps.

The weighing step is done by updating the self-normalized weights given in GENPF algorithm. We define \( w_{d,i} \) and \( r_{d,i} \) as the weights and rewards obtained at step \( d \) for state sequence \( i \) from GENPF simulation. With our recursive definition of the empirical weights, we obtain the full weight of each state sequence \( i \) for a fixed observation sequence \( j \):

\[
w_{d,i} = w_{d-1,i} \cdot Z(o_{d,j} | a_{d,i}, s_{d,i}) \propto Z_{1:d}^i. \tag{33}
\]

Realizing that the marginal observation probability is independent of indexing by \( i \), we show that \( \hat{\mu}_{\theta_d}[f] \) is an SN estimator of \( \mathbb{E}_{s \sim \theta_d}[f(s)] \):

\[
\hat{\mu}_{\theta_d}[f] = \frac{\sum_{C=1}^{C} (Z_{1:d}^i)f(s_{d,i})}{\sum_{C=1}^{C} (Z_{1:d}^i)} = \frac{\sum_{C=1}^{C} (Z_{1:d}^i)f(s_{d,i})}{\sum_{C=1}^{C} (Z_{1:d}^i)} \tag{34}
\]

Since \( \{s_n\}_1, \ldots, \{s_n\}_C \) are i.i.d.r.v. sequences of depth \( d \), and \( f \) is a bounded function, we can apply the SN concentration bound in Theorem 1 to obtain the concentration inequality. Since \( d_{\infty}(\mathcal{P}^d || \mathcal{Q}^d \) is bounded by \( d_{\max} \) a.s., we can bound the resulting \( t_d(\lambda, C) \) by \( t_{\max}(\lambda, C) \) a.s.:

\[
t_d(\lambda, C) = \frac{\lambda}{f_{\max}d_{\infty}(\mathcal{P}^d || \mathcal{Q}^d)} - \frac{1}{\sqrt{C}} \geq \frac{\lambda}{f_{\max}d_{\max} - 1} \equiv t_{\max}(\lambda, C) \tag{36}
\]

This means that for all \( d \), we can bound \( t_d(\lambda, C) \geq t_{\max}(\lambda, C) \). Thus, bounding the concentration inequality probability with \( t_{\max}(\lambda, C) \) for any step \( d \) is justified when we prove Lemma 2 later. This probabilistic bound holds for any choice of \( \{o_{n,j}\} \), where \( \{o_{n,j}\} \) could be a sequence of random variables correlated with any elements of \( \{s_{n,i}\} \). Thus, for any \( \{o_{n,j}\} \),

\[
|\mathbb{E}_{s \sim \theta_d}[f(s)] - \hat{\mu}_{\theta_d}[f]| \leq \lambda \tag{37}
\]

holds with probability at least \( 1 - 3\exp(-C \cdot t_{\max}^2(\lambda, C)) \).
Appendix C. Proof of Lemma 2 (Continued) - Sparse Sampling-ω Q-Value Coupled Convergence

Lemma 2 (Sparse Sampling-ω Estimator Q-Value Coupled Convergence). For all \( d = 0, \ldots, D - 1 \) and \( a \), the following bounds hold with probability at least \( 1 - 6/|A|(4|C|)^2 \exp(-C \cdot \hat{t}^2) \):

\[
\begin{align*}
|Q^*_p(b_d, a) - \hat{Q}^*_\omega(b_d, a)| & \leq \alpha_d, \quad \alpha_d = \lambda + \gamma \alpha_{d+1}, \quad \alpha_{D-1} = \lambda, \quad (38) \\
|Q^*_m(b_d, a) - \hat{Q}^*_\omega(b_d, a)| & \leq \beta_d, \quad \beta_d = \gamma(\lambda + \beta_{d+1}), \quad \beta_{D-1} = 0, \quad (39) \\
t_{\max}(\lambda, C) & = \frac{\lambda}{4V_{\max}d_{\max}} - \frac{1}{\sqrt{C}}, \quad \hat{t} = \min\{t_{\max}, \lambda/4 \sqrt{2V_{\max}}\} \quad (40)
\end{align*}
\]

Before we proceed with the proof, note that in our definition of \( t_{\max} \), we set the maximum of the \( f_{\max} \) to be equal to \( 4V_{\max} \). While this may seem very conservative to bound most reasonable functions resulting from reward and value estimation with 4 times the \( V_{\max} \), it serves to uniformly bound the probability for each of the SN estimator terms with convenient coefficients. Furthermore, individual concentration bounds may be adjusted to account for this generous upper bound by multiplying a factor in front of \( \lambda \).

POMDP Value Convergence: We split the difference between the SN estimator and \( Q^*_p \) into two terms, the reward estimation error (A) and the next-step value estimation error (B):

\[
|Q^*_p(b_d, a) - \hat{Q}^*_\omega(b_d, a)| \leq \left| \mathbb{E}[R(s_d, a) \mid b_d] - \sum_{i=1}^{C} W_{d,i} R_{d,i} \right| + \gamma \mathbb{E}[V_{d+1}\omega(bao) \mid b_d] - \frac{1}{C} \sum_{i=1}^{C} \hat{V}^*_{d+1}(b_d|i) \quad (41)
\]

Here, the \( I_i \) notation represents that random variables \( I_i \) are sampled \( C \) times from the finite discrete distribution \( p_{w,d} \) with probability mass \( p_{w,d}(I = i) = (w_{d,i}/\sum_j w_{d,j}) \), and particle belief state \( \bar{b}_{d+1|i} \) is updated by an observation generated from \( s_{d,i} \). This reflects the fact that GENPF randomly selects a state particle \( s_\omega \) with probability \( w_{d,i}/\sum_j w_{d,j} \) \( C \) times independently to generate a new observation for the next step particle belief state. Similarly, a particle belief state \( \bar{b}_{d+1|i} \) is updated by an observation generated from \( s_{d,i} \), which is a notation we will use to represent beliefs that are generated through iterating upon each state particle \( s_{d,i} \).

To prove the base case \( d = D - 1 \), we note that we only need to bound the first term (A) since \( d = D - 1 \) corresponds to the leaf node of Sparse Sampling-ω tree and no further next step value estimation is performed:

\[
|Q^*_p(b_{D-1}, a) - \hat{Q}^*_\omega(b_{D-1}, a)| \leq \left| \mathbb{E}[R(s_{D-1}, a) \mid b_{D-1}] - \sum_{i=1}^{C} W_{D-1,i} R_{D-1,i} \right|. \quad (42)
\]

This term is simply a particle likelihood weighted average estimation term where the function is \( R(\cdot, a) \), and does not need any inductive step. Below, we will show how to bound both terms (A) and (B), so the base case proof naturally follows from the proof of concentration bound for (A).

For (A), we use the particle likelihood SN concentration bound in Lemma 1 to obtain the bound \( \frac{R_{\max}}{4V_{\max}} \lambda \); rather than bounding \( R \) with \( 4V_{\max} \) in this step, we instead bound \( R \) with \( R_{\max} \) and then augment \( \lambda \) to \( \frac{R_{\max}}{4V_{\max}} \lambda \) in order to obtain the same uniform \( t_{\max} \) factor as the other steps. This choice
of bound is made to effectively combine the $\lambda$ terms when we add (A) and (B). This also covers the base case since $\alpha_{d-1} = \lambda \geq \frac{R_{\text{max}}}{V_{\text{max}}} \lambda$.

For (B), we use the triangle inequality repeatedly to separate it into four terms; (1) the importance sampling error bounded by $\lambda / 4$, (2) the Monte Carlo weighted sum approximation error bounded by $\lambda / 4$, (3) the Monte Carlo next-step integral approximation error bounded by $\lambda / 2$, and (4) the function estimation error bounded by $\alpha_{d+1}$:

$$
\mathbb{E}[V_{d+1}^*(bao) \mid b_d] - \frac{1}{C} \sum_{i=1}^{C} \hat{V}_{d+1}^*(\hat{b}_d) - \frac{1}{C} \sum_{i=1}^{C} \hat{V}_{d+1}^*(\hat{b}_d) \leq
$$

$$
\left| \mathbb{E}[V_{d+1}^*(b_d) \mid b_d] - \frac{1}{C} \sum_{i=1}^{C} V_{d+1}^*(b_d, a)[i] \right| + \sum_{i=1}^{C} w_{d,i} \left| V_{d+1}^*(b_d, a)[i] \right| - \frac{1}{C} \sum_{i=1}^{C} V_{d+1}^*(b_d, a)[i]
$$

(1) Importance sampling error

$$
+ \frac{1}{C} \sum_{i=1}^{C} V_{d+1}^*(b_d, a)[i] - \frac{1}{C} \sum_{i=1}^{C} V_{d+1}^*(b_d, a)[i]
$$

(2) MC weighted sum approximation error

$$
+ \frac{1}{C} \sum_{i=1}^{C} V_{d+1}^*(b_d, a)[i] - \frac{1}{C} \sum_{i=1}^{C} V_{d+1}^*(b_d, a)[i]
$$

(3) MC next-step integral approximation error

$$
+ \frac{1}{C} \sum_{i=1}^{C} \hat{V}_{d+1}^*(\hat{b}_d) - \frac{1}{C} \sum_{i=1}^{C} \hat{V}_{d+1}^*(\hat{b}_d)
$$

(4) Function estimation error

The following subsections justify how each error term is bounded.

(1) **Importance Sampling Error**: Before we analyze the first term, note that the conditional expectation of the optimal value function at step $d + 1$ given $b_d, a$ is calculated by the following, where we introduce $V_{d+1}^*(b_d, a, s_{d,i}) \equiv V_{d+1}^*(b_d, a)[i]$ as a shorthand for the next-step integration over $(s_{d+1}, o)$ conditioned on $(b_d, a, s_{d,i})$. Once again, we denote $[i]$ to indicate that $s_{d,i}$ was the particle chosen to generate the observation $o$, and if we are conditioning on a generic particle $s_d$, then we simply denote all the variables $V_{d+1}^*(b_d, a, s_d)$:

$$
V_{d+1}^*(b_d, a)[i] \equiv \int_{s} \int_{o} V_{d+1}^*(b_d, a) \mathcal{Z}(o \mid a, s_{d+1}) T(s_{d+1} \mid s_{d,i}, a) ds_{d+1} do
$$

(45)

$$
\mathbb{E}[V_{d+1}^*(b_d) \mid b_d] = \int_{s} \int_{o} V_{d+1}^*(b_d, a) \mathcal{Z}(o \mid a, s_{d+1}) T(s_{d+1} \mid s_{d,i}) b_d \cdot ds_{d+1} d o
$$

(46)

$$
= \int_{s} V_{d+1}^*(b_d, a, s_d) b_d \cdot ds_d
$$

(47)

$$
= \int_{s_{d+1}} V_{d+1}^*(b_d, a, s_d) \mathcal{Z}(s_{d+1} \mid s_{d,i}) T(s_{d+1} \mid s_{d,i}) b_d \cdot ds_{d+1}
$$

(48)

Noting that the term (1) is then the difference between the SN estimator and the conditional expectation, and that $\|V_{d+1}^*\|_m \leq V_{\text{max}}$, we can apply the SN inequality for the second time in Lemma 2 to bound it by the augmented $\lambda / 4$. Thus, with our definition of $t_{\text{max}}$, the bound holds with probability at least $1 - 3 \exp(-C \cdot t_{\text{max}}^2(\lambda, C))$.

(2) **Monte Carlo Weighted Sum Approximation Error**: The second term is the error resulting from estimating the sum with a Monte Carlo sum, which can be bounded by a Hoeffding-type
random variable is the probability of the associated event. By denoting the difference as statement by applying the Tower property, and noting that the expectation of an indicator \( I \) does not depend on the specific values of these weights nor the particle belief sets, since the previous calculation was done by conditioning on \( \lambda \) with probability at least \( 1 - 2\exp(-C\lambda^2/2V_{\text{max}}^2) \) for an arbitrary fixed set of \( \{s_{d,i}, w_{d,i}\}, b_d, a \):

\[
\left| \sum_{i=1}^{C} w_{d,i} \frac{V_{d+1}^*(b_d, a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} - \frac{1}{C} \sum_{i=1}^{C} V_{d+1}(b_d, a)^{[i]} \right| \leq \lambda. \tag{51}
\]

The previous calculation was done by conditioning on \( \{s_{d,i}, w_{d,i}\}, b_d, a \). However, this bound does not depend on the specific values of these weights nor the particle belief sets, since Hoeffding bound only takes advantage of the fact that the random variables \( I_i \) are sampled i.i.d. and the corresponding \( V(I_i) \) are bounded. Thus, we can revert this back into a general statement by applying the Tower property, and noting that the expectation of an indicator random variable is the probability of the associated event. By denoting the difference as \( \Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \), we obtain the unconditional Hoeffding-type bound:

\[
\mathbb{P} \left\{ \Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \leq \lambda \right\} = \mathbb{E} \left[ \mathbb{1}_{\{\Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \leq \lambda\}} \right] \\
= \mathbb{E} \left[ \mathbb{E} \left[ \mathbb{1}_{\{\Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \leq \lambda\}} \mid \{s_{d,i}, w_{d,i}\}, b_d, a \right] \right] \\
= \mathbb{E} \left[ \mathbb{P} \left\{ \Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \leq \lambda \mid \{s_{d,i}, w_{d,i}\}, b_d, a \right\} \right] \\
\geq \mathbb{E} \left[ 1 - 2\exp(-C\lambda^2/2V_{\text{max}}^2) \right] \\
= 1 - 2\exp(-C\lambda^2/2V_{\text{max}}^2). \tag{55}
\]

Here, we use the factor augmentation once again to choose \( \lambda/4 \) such that the absolute difference is bounded by \( \lambda/4 \) with probability at least \( 1 - 2\exp(-C\lambda^2/32V_{\text{max}}^2) \), which gets us our desired result:

\[
\left| \sum_{i=1}^{C} w_{d,i} \frac{V_{d+1}^*(b_d, a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} - \frac{1}{C} \sum_{i=1}^{C} V_{d+1}(b_d, a)^{[i]} \right| \leq \lambda/4. \tag{57}
\]
(3) **Monte Carlo Next-Step Integral Approximation Error:** The third term can be thought of as Monte Carlo next-step integral approximation error. To estimate $V^*_{d+1}(b_d,a)^{[l]}$, we can simply use the quantity $V^*_{d+1}(b_dao^{[l]})$, as the random vector $(s_{d+1},I_d,o_d)$ is jointly generated using $G$ according to the correct probability $Z(o'|a,s_{d+1})T(s_{d+1}|s_{d},a)$ given $s_d,a$ in the simulation realized in the tree. Consequently, the quantity $V^*_{d+1}(b_dao^{[l]})$ for a given $(s_{d+1},I_d,a)$ is an unbiased 1-sample MC estimate of $V^*_{d+1}(b_d,a)^{[l]}$. We define the difference between these two quantities as $\Delta_{d+1}$, which is implicitly a function of random variables $(s_{d+1},I_d,o_d)$:

$$\Delta_{d+1}(b_d,a)^{[l]} \equiv V^*_{d+1}(b_d,a)^{[l]} - V^*_{d+1}(b_dao^{[l]})$$

Then, we note that $\|\Delta_{d+1}\|_\infty \leq 2V_{\text{max}}$ and $\mathbb{E}\Delta_{d+1} = 0$ by the Tower property conditioning on $(s_{d+1},I_d,a)$ (which is implicitly conditioning on $I$, but this does not matter greatly as everything cancels out) and integrating over $(s_{d+1},I_d,o_d)$ first, which holds for any choice of well-behaved sampling distributions on $(s_{0:d})$. Using this fact, we can then consider this term as a Monte Carlo estimator for the bias $\mathbb{E}\Delta_{d+1} = 0$, and use another Hoeffding bound. Since $\|\Delta_{d+1}\|_\infty \leq 2V_{\text{max}}$, our $\Delta$ factor is then augmented by $1/2$ to once again obtain probability at least $1 - 2\exp(-C\lambda^2/2V_{\text{max}}^2)$:

$$\left| \frac{1}{C} \sum_{i=1}^{C} V^*_{d+1}(b_d,a)^{[l]} - \frac{1}{C} \sum_{i=1}^{C} V^*_{d+1}(b_dao^{[l]}) \right| = \left| \frac{1}{C} \sum_{i=1}^{C} (V^*_{d+1}(b_d,a)^{[l]} - V^*_{d+1}(b_dao^{[l]})) - 0 \right|
= \left| \frac{1}{C} \sum_{i=1}^{C} \Delta_{d+1}(b_d,a)^{[l]} - \mathbb{E}\Delta_{d+1} \right| \leq \frac{\lambda}{2}.$$ 

(4) **Function Estimation Error:** The fourth term is bounded by the inductive hypothesis, since each $i$-th absolute difference of the $Q$-function and its estimate at step $d + 1$, and furthermore the value function and its estimate at step $d + 1$, are all bounded by $\alpha_{d+1}$.

Thus, each of the error terms are bound by (A) $\leq \frac{R_{\text{max}}}{4V_{\text{max}}} \lambda$ and (B) $\leq \frac{1}{4} \lambda + \frac{1}{4} \lambda + \frac{1}{2} \lambda + \alpha_{d+1}$, which uses the SN concentration bound 2 times and Hoeffding bound 2 times. Combining (A) and (B), we can obtain the desired bound:

$$|Q^*_d(b_d,a) - \bar{Q}_d(b_d,a)| \leq \frac{R_{\text{max}}}{4V_{\text{max}}} \lambda + \gamma \left[ \frac{1}{4} \lambda + \frac{1}{4} \lambda + \frac{1}{2} \lambda + \alpha_{d+1} \right]$$

$$\leq \frac{1 - \gamma}{4} \lambda + \gamma \left[ \frac{1}{4} \lambda + \frac{3}{4} \gamma + \alpha_{d+1} \right]$$

$$= \lambda + \gamma \alpha_{d+1} = \alpha_d.$$ 

Now, we obtain the worst case union bound. We now derive the worst case union bound probability. First, we want to ensure that the SN concentration inequality holds with probability $1 - 3\exp(-C \cdot t_{\text{max}}^2(\lambda, C))$ whenever it is used at any given step $d$ and action $a$. Similarly, we also want to ensure that the Hoeffding-type inequality holds with probability at least $1 - 2\exp(-C\lambda^2/2V_{\text{max}}^2)$ whenever it is used at any given step $d$ and action $a$. This means we can bound the worst case probability of using either bound by

$$\max(3\exp(-C \cdot t_{\text{max}}^2(\lambda, C)), 2\exp(-C\lambda^2/2V_{\text{max}}^2))$$

$$\leq 3\exp(-C \cdot t_{\text{max}}^2(\lambda, C)) + 2\exp(-C\lambda^2/2V_{\text{max}}^2)$$

$$\leq 5\exp(-C \cdot t^2).$$
Furthermore, we multiply the worst-case union bound factor \( (4|A|C)^D \), since we want the function estimates to be within their respective concentration bounds for all the actions \(|A|\) and child nodes \(C\) at each step \(d = 0, \ldots, D - 1\), for the 2 times we use SN concentration bound and 2 times we use the double-sided Hoeffding-type bound in the induction step. We once again multiply the final probability by \(|A|\) to account for the root node \(Q\)-value estimates also satisfying their respective concentration bounds for all actions. Thus, the worst case union bound probability of all bad events is bounded by probability \( 5|A|(4|A|C)^D \exp(-C \cdot \bar{r}^2) \). Therefore, we have shown that the concentration bounds for both the particle likelihood SN estimator and Monte Carlo estimator components converge with probability at least \(1 - 5|A|(4|A|C)^D \exp(-C \cdot \bar{r}^2)\) for all levels \(d\):

\[
|Q^*_p(b_d, a) - \hat{Q}^*_d(b_d, a)| \leq \alpha_d.
\] (66)

**PB-MDP Value Convergence:** Once again, we split the difference between the SN estimator and the \(Q^*_p\) function into two terms, the reward estimation error \((A)\) and the next-step value estimation error \((B)\):

\[
|Q^*_p(b_d, a) - \hat{Q}^*_d(b_d, a)| \leq |\rho(b_d, a) - \rho(\bar{b}_d, a)| + \gamma \mathbb{E}_{M_p}[V^*_{M_p,d+1}(\bar{b}_{d+1}) | \bar{b}_d, a] - \frac{1}{C} \sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(b_{d+1}^{|l|})|.
\] (A) = 0

\[
\mathbb{E}_{M_p}[V^*_{M_p,d+1}(\bar{b}_{d+1}) | \bar{b}_d, a] - \frac{1}{C} \sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(b_{d+1}^{|l|})|.
\] (B)

\[
|Q^*_p(b_d, a) - \hat{Q}^*_d(b_d, a)| \leq |\rho(b_d, a) - \rho(\bar{b}_d, a)| + \gamma \mathbb{E}_{M_p}[V^*_{M_p,d+1}(\bar{b}_{d+1}) | \bar{b}_d, a] - \frac{1}{C} \sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(b_{d+1}^{|l|})|.
\] (67)

Since our particle belief MDP induces no reward estimation error, the term \((A)\) is always 0 and proving the base case \(d = D - 1\) is trivial as \((A)\) and \((B)\) are both 0.

We now prove that the difference \((B)\) is bounded for all \(d = 0, \ldots, D - 1\). We use the triangle inequality repeatedly to separate it into two terms; (1) the MC transition approximation error bounded by \(\lambda\), and (2) the function approximation error bounded by \(\beta_{d+1}\):

\[
\mathbb{E}_{M_p}[V^*_{M_p,d+1}(\bar{b}_{d+1}) | \bar{b}_d, a] - \frac{1}{C} \sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(b_{d+1}^{|l|})|.
\] (B)

\[
\leq \mathbb{E}_{M_p}[V^*_{M_p,d+1}(\bar{b}_{d+1}) | \bar{b}_d, a] - \frac{1}{C} \sum_{i=1}^{C} V^*_{M_p,d+1}(b_{d+1}^{|l|})| + \frac{1}{C} \sum_{i=1}^{C} V^*_{M_p,d+1}(b_{d+1}^{|l|}) - \frac{1}{C} \sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(b_{d+1}^{|l|})|.
\] (1) MC transition approximation error

\[
\leq \lambda
\] (1)

\[
+ \beta_{d+1}.
\] (2)

We justify how each error term is bounded.

**1) MC Transition Approximation Error:** The Monte Carlo summation over the next step particle belief state samples \(\{b_{d+1}^{|l|}\}\) given \((b_d, a)\) is essentially approximating the integration over the transition density \(\tau(\bar{b}_d, a)\). Since the value function and its estimate are both bounded by \(V_{\max}\), we can invoke Hoeffding bound once again to obtain the following exponential probabilistic bound on the difference:

\[
\mathbb{P} \left\{ \mathbb{E}_{M_p}[V^*_{M_p,d+1}(\bar{b}_{d+1}) | \bar{b}_d, a] - \frac{1}{C} \sum_{i=1}^{C} V^*_{M_p,d+1}(b_{d+1}^{|l|}) \leq \lambda \right\} \geq 1 - 2 \exp(-C\lambda^2/2V_{\max}^2).
\] (70)
Function Approximation Error: The second term is bounded by the inductive hypothesis, since each \(i\)-th absolute difference of the \(Q\)-function and its estimate at step \(d + 1\), and furthermore the value function and its estimate at step \(d + 1\), are all bounded by \(\beta_{d+1}\).

By applying similar logic of ensuring that every particle belief state node and action pairs can satisfy the concentration inequality, we note that the particle belief MDP approximation concentration bound is satisfied with probability at least \(1 - \left|A\right| \cdot \left|A\right| C \cdot D \exp(-C \cdot \tilde{t}^2)\). Thus, since (A) is 0 and (B) is bounded by \(\lambda + \beta_{d+1}\), the \(Q\) value estimation error with respect to \(M_P\) is bounded as desired:

\[
|Q^*_{M_P,d}(\bar{b}_d,a) - \hat{Q}^*_{\omega,d}(\bar{b}_d,a)| \leq \gamma (\lambda + \beta_{d+1}) = \beta_d.
\]

Combining both concentration bounds: In order to enable the two concentration inequalities to happen simultaneously, we bound the worst case union probability by using the definition of \(\overline{t}\) and combining the upper bounding terms together:

\[
\begin{align*}
5\left|A\right|(4\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2) &+ \left|A\right|(\left|A\right|C)^D \exp(-C \lambda^2 / 2V^2_{\text{max}}) \leq 5\left|A\right|(4\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2) &+ \left|A\right|(\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2) \\
&\leq 5\left|A\right|(4\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2) &+ \left|A\right|(4\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2) \\
&= 6\left|A\right|(4\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2).
\end{align*}
\]

Therefore, we conclude that the \(Q\) value concentration inequalities for both POMDP approximation error and particle belief approximation error are bounded by \(\alpha_d, \beta_d\) at every node, respectively, with probability at least \(1 - 6\left|A\right|(4\left|A\right|C)^D \exp(-C \cdot \tilde{t}^2)\).
Appendix D. Proof of Theorem 2 - Sparse Sampling-ω Coupled Optimality

We reiterate the conditions and Theorem 2 below:

(i) $S$ and $O$ are continuous spaces, and the action space has a finite number of elements, $|A| < +\infty$.

(ii) For any observation sequence $\{o_n\}_{n=1}^d$, the densities $Z, T, b_0$ are chosen such that the Rényi divergence of the target distribution $P^d$ and sampling distribution $Q^d$ (Eqs. (30) and (31)) is bounded above by $d_{\infty}^\max$ for all $d = 0, \ldots, D - 1$:

$$d_{\infty}(P^d||Q^d) = \text{ess sup}_{x \sim Q^d} \frac{w_{P^d}(x)}{w_{Q^d}(x)} \leq d_{\infty}^\max$$

(iii) The reward function $R$ is bounded by a finite constant $R_{\max}$, and hence the value function is bounded by $V_{\max} \equiv R_{\max} / (1 - \gamma)$.

(iv) We can sample from the generating function $G$ and evaluate the observation probability density $Z$.

(v) The POMDP terminates after no more than $D < \infty$ steps.

**Theorem 2** (Sparse Sampling-ω Coupled Optimality). Suppose conditions (i)-(v) are satisfied. Then, for any desired policy optimality $\varepsilon > 0$, choosing constants $C, \lambda, \delta$ that satisfy:

$$\lambda = \varepsilon(1 - \gamma)^2 / 8,$$

$$\delta = \lambda / (V_{\max}D(1 - \gamma)^2),$$

$$C = \max \left\{ \left( 4V_{\max}d_{\infty}^\max \right)^2 \frac{64V_{\max}^2}{\lambda^2} \left( D \log \frac{24|A|^{\frac{D}{2}}V_{\max}^2D}{\lambda^2} + \log \frac{1}{\delta} \right) \right\},$$

the $Q$-function estimates $\hat{Q}_{\omega,d}(\bar{b}_d, a)$ obtained for all depths $d = 0, \ldots, D - 1$ and all actions $a$ are jointly near-optimal with respect to $Q^d_{P,\omega}$ and $Q^d_{M,\omega}$ with probability at least $1 - \delta$:

$$|Q^d_{P,\omega}(b_d, a) - \hat{Q}_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1 - \gamma},$$

$$|Q^d_{M,\omega}(\bar{b}_d, a) - \hat{Q}_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1 - \gamma}.$$

**Proof.** This proof has two parts. First, we show that the choice of $C$ is valid given the assumptions in Lemma 2. Then, we use Lemmas 2 and 3A to prove the Q-value estimate claim.

The conditions necessary for $C$ from Lemma 2 are the following:

$$t_{\max}(\lambda, C) = \frac{\lambda}{4V_{\max}d_{\infty}^\max} > 0$$

$$\delta \geq 6|A|(4|A|C)^D \exp(-C \cdot \bar{t}^2)$$

$$\bar{t}_{\max}(\lambda, C) \equiv \max \left\{ \bar{t}_{\max}(\lambda, C), \lambda / 4\sqrt{2}V_{\max} \right\}$$
Note that the constraint on $t_{\text{max}}$ implies that the following must be true:

$$\frac{\lambda}{4V_{\text{max}}d_{\text{max}}} - \frac{1}{\sqrt{C}} > 0 \implies C > \left(\frac{4V_{\text{max}}d_{\text{max}}}{\lambda}\right)^2,$$

which gives us the first option of $C$ in the maximum.

For the next option of $C$, we show that substituting the formula yields condition Eq. (83). We note that due to the definition of $\tilde{t}_{\text{max}}$, the following is true:

$$\tilde{t}_{\text{max}}(\lambda, C) \geq \frac{\lambda}{4\sqrt{2}V_{\text{max}}}.$$

Let us denote $T \equiv \left(\frac{\lambda}{4\sqrt{2}V_{\text{max}}}\right)^2$ for convenience. Then, since $T$ is upper-bounded by $\tilde{t}_{\text{max}}$,

$$6|A|(4|A|C)^D \exp(-C \cdot \tilde{t}^2) \leq 6|A|(4|A|C)^D \exp(-C \cdot T)$$

$$\leq (24|A|^{D+1})^D \exp(-C \cdot T)$$

Consequently, if we show that Eq. (88) is bounded by $\delta$, then we automatically show that the original quantity is bounded by $\delta$ as well. By defining $X \equiv 24|A|^{D+1}$, we want to show that this simplified formula is bounded above by $\delta$:

$$\delta \geq (X \cdot C)^D \exp(-C \cdot T).$$

We will show that our second option of $C$ satisfies the following, where the simplified formula equals:

$$\frac{64V_{\text{max}}^2}{\lambda^2} \left(D \log \frac{24|A|^{D+1}V_{\text{max}}D}{\lambda^2} + \log \frac{1}{\delta}\right) \implies \frac{2}{T} \left(D \log \frac{XD}{T} + \log \frac{1}{\delta}\right)$$

Substituting in the second option of $C$:

$$(X \cdot C)^D \exp(-C \cdot T) = \delta^2 \left(\frac{2XD}{T} \log \frac{XD}{T} + \frac{2X}{T} \log \frac{1}{\delta}\right)^D$$

$$= \delta^2 \left(\frac{XD}{T} \log \left(\frac{XD}{T}\right)^2 + \frac{X}{T} \log \left(\frac{1}{\delta}\right)^{2/D}\right)^D$$

$$= \delta^2 \left(\log \left(\frac{XD}{T^{3/D}}\right)^2\right)^D = \delta \left(\frac{XD}{T^{3/D}}\right)^D$$

Note that the function $f(x) = \log x^2/x$ is less than 1 for $x > 0$ (in fact, the maximum value of $f(x)$ is exactly $2/e$, attained by setting $x = e$). This means that the quantity inside the parentheses is less than 1, which lets us obtain our desired result

$$(X \cdot C)^D \exp(-C \cdot T) \leq \delta.$$
Therefore, each of our option of $C$ satisfies the respective conditions, and taking the maximum of the options will yield valid results for both inequality constraints:

$$C = \max \left\{ \left( \frac{4V_{\text{max}}d_{\text{max}}}{\lambda} \right)^2, \frac{64V_{\text{max}}^2}{A^2} \left( D \log \frac{24|A|d_{\text{max}}V_{\text{max}}D}{\lambda^2} + \log \frac{1}{\delta} \right) \right\}. \quad (95)$$

This concludes the first part of the proof; we have shown that $C$ is a valid choice. Next, we prove the value bounds.

With our choice of $C$, from Lemma [2] the error in estimating $Q^*$ with our Sparse Sampling-$\omega$ policy is bounded by $\alpha_d$ for all $d, a$ with probability at least $1 - \delta$. Since $\alpha_d \leq \alpha_0$, the following holds for all $d = 0, \ldots, D - 1$ with probability at least $1 - \delta$ through Lemmas [1] and [2]:

$$|Q_{P,d}(b_d, a) - \hat{Q}_{\omega}(\tilde{b}_d, a)| \leq \alpha_0 \leq \sum_{d=0}^{D-1} \gamma^d \lambda \leq \frac{\lambda}{1 - \gamma}, \quad (96)$$

$$|Q_{M,d}(\tilde{b}_d, a) - \hat{Q}_{\omega}(\tilde{b}_d, a)| \leq \beta_0 \leq \sum_{d=1}^{D} \gamma^d \lambda \leq \frac{\lambda}{1 - \gamma}. \quad (97)$$

\[\square\]
Appendix E. Proof of Theorem 4 - Particle Belief MDP Approximate Policy Convergence

Before we prove Theorem 4, we first prove the following lemma, which is an adaptation of Kearns et al. (2002) and Singh and Yee (1994) for belief states $b$.

**Lemma 3A.** Consider a POMDP with a finite horizon of $D$ steps and policy $\pi(b) = \arg\max_a \hat{Q}(b,a)$ where $\hat{Q}$ is a stochastic value function approximator with errors bounded by a positive constant $\xi$: $|Q^*(b,a) - \hat{Q}(b,a)| \leq \xi$. Let $V^\pi(b_0)$ denote the value of executing $\pi$ starting at belief $b_0$ with an exact Bayesian belief update, $b_{t+1} = b_{t\alpha}$, between each call to the policy. Then

$$V^*(b_0) - V^\pi(b_0) \leq \frac{2\xi}{1-\gamma}. \quad (98)$$

**Proof.** First, note that if an action is chosen by $\pi$, it must appear better than $\pi^*$ according to $\hat{Q}$, i.e. $\hat{Q}(b,\pi(b)) \geq \hat{Q}(b,\pi^*(b))$. The worst case is when $\hat{Q}(b,\pi(b)) = Q^*(b,\pi(b)) + \xi$ and $\hat{Q}(b,\pi^*(b)) = Q^*(b,\pi^*(b)) - \xi$. Thus, for any $t$, we have the bound

$$Q^*(b_t, \pi^*(b_t)) - E[Q^*(b_t, \pi(b_t))] \leq 2\xi. \quad (99)$$

Next, we prove that $V^*(b_t) - V^\pi(b_t) \leq \sum_{d=t}^{D-1} \gamma^{d-t}2\xi$ using induction from $t = D - 1$ to $t = 0$. We verify the base case, $t = D - 1$, by observing that both $Q^\pi(b_{D-1}, \pi(b_{D-1}))$ and $Q^*(b_{D-1}, \pi(b_{D-1}))$ are equal to $R(b_{D-1}, \pi(b_{D-1}))$ since no further reward can be accumulated and using Eq. (99):

$$V^*(b_{D-1}) - V^\pi(b_{D-1}) = Q^*(b_{D-1}, \pi^*(b_{D-1})) - E[Q^\pi(b_{D-1}, \pi(b_{D-1}))] \leq 2\xi. \quad (100)$$

$$= Q^*(b_{D-1}, \pi^*(b_{D-1})) - E[Q^*(b_{D-1}, \pi(b_{D-1}))] \leq 2\xi. \quad (101)$$

The inductive step is verified by subtracting and adding $E[Q^*(b, \pi(b))]$, using the bound in Eq. (99), and applying the inductive hypothesis:

$$V^*(b_t) - V^\pi(b_t) = Q^*(b, \pi^*(b)) - E[Q^\pi(b, \pi(b))]$$

$$= Q^*(b, \pi^*(b)) - E[Q^*(b, \pi(b))] + E[Q^*(b, \pi(b))] - E[Q^\pi(b, \pi(b))] \quad (103)$$

$$\leq 2\xi + E[Q^*(b, \pi(b))] - E[Q^\pi(b, \pi(b))] \quad (104)$$

$$= 2\xi + E[R(b, \pi(b))] + \gamma E[V^\pi(b_{t+1})] - E[R(b, \pi(b))] - \gamma E[V^\pi(b_{t+1})] \quad (105)$$

$$= 2\xi + \gamma E[V^\pi(b_{t+1})] - V^\pi(b_{t+1})] \quad (106)$$

$$= 2\xi + \gamma \sum_{d=t+1}^{D-1} \gamma^{d-t}2\xi = \sum_{d=t}^{D-1} \gamma^{d-t}2\xi. \quad (107)$$

Now, by applying the result above to $t = 0$, we prove the lemma:

$$V^*(b_0) - V^\pi(b_0) \leq \sum_{d=0}^{D-1} \gamma^d2\xi \leq \frac{2\xi}{1-\gamma}. \quad (108)$$

$\square$

39
Theorem 4 (Particle Belief MDP Approximate Policy Convergence). In addition to regularity conditions for particle belief MDP and the MDP planning algorithm $A$, assume that the closed-loop POMDP Bayesian belief update step is exact. Then, for any $\varepsilon > 0$, we can choose a $C$ such that the value obtained by planning with $A$ in the PB-MDP is within $\varepsilon$ of the optimal POMDP value function at $b_0$:

$$V_P^*(b_0) - V_{MP}^A(b_0) \leq \varepsilon.$$  \hfill (109)

Proof. During policy execution, we create a new independent tree and choose an action based on the estimated Q-values. At each of the $D$ steps of the POMDP, there is at most $\delta$ probability that $|Q_{P,0}(b_0, a) - \hat{Q}_{MP,0}(\bar{b}_0, a)| > \frac{2\lambda}{1-\gamma} + \varepsilon_A$. We further choose $C', \delta', \lambda'$ such that $\frac{2\lambda'}{1-\gamma} = \frac{2\lambda}{1-\gamma} + \varepsilon_A$.

Thus, with probability at least $1 - D\delta'$, we execute a policy that meets the assumptions of Lemma 3A with $\xi = \frac{2\lambda'}{1-\gamma}$, and hence by Lemma 3A, the difference between the optimal value and the average accumulated reward for this case is at most $\frac{4\lambda'}{(1-\gamma)^2}$. In the other case, which occurs with at most probability $D\delta'$, an arbitrarily bad policy can be executed, resulting in an accumulated reward difference of up to $2V_{\text{max}}$ from the optimal policy. Combining these two cases and using the definition of $\delta'$ in Eq. (29), we have

$$V_P^*(b_0) - V_{MP}^A(b_0) \leq (1 - D\delta') \frac{4\lambda'}{(1-\gamma)^2} + D\delta'2V_{\text{max}}$$  \hfill (110)

$$\leq \frac{4\lambda'}{(1-\gamma)^2} + \frac{4\lambda'}{(1-\gamma)^2}$$  \hfill (111)

$$= \varepsilon.$$  \hfill (112)

$\square$
### Appendix F. Experiment Details

|                        | $c$ | $k_a$ | $\alpha_d$ | $k_o$ | $\alpha_o$ | $m_{\text{min}}$ | $\delta$ | Depth |
|------------------------|-----|-------|------------|-------|------------|-------------------|----------|-------|
| **Laser Tag (D, D, D)** |     |       |            |       |            |                   |          |       |
| Sparse-PFT             | 26  | -     | 4          | -     | -          | 50                |          |       |
| PFT-DPW                | 26  | -     | 4          | 1/35  | -          | 50                |          |       |
| POMCPOW                | 26  | -     | 4          | 1/35  | -          | 50                |          |       |
| AdaOPS                 | -   | -     | -          | -     | -          | 30                | 0.1      | 90    |
| POMCP                  | 26  | -     | -          | -     | -          | -                 |          |       |
| QMDP                   | -   | -     | -          | -     | -          | -                 |          |       |
| **Light Dark (D, D, C)**|     |       |            |       |            |                   |          |       |
| Sparse-PFT             | 100 | -     | 4          | -     | -          | 20                |          |       |
| PFT-DPW                | 100 | -     | 4          | 1/10  | -          | 20                |          |       |
| POMCPOW                | 90  | -     | 5          | 1/15  | -          | 20                |          |       |
| AdaOPS                 | -   | -     | -          | -     | -          | 30                | 0.1      | 90    |
| POMCP                  | 83  | -     | -          | -     | -          | -                 |          |       |
| QMDP                   | -   | -     | -          | -     | -          | -                 |          |       |
| **Sub Hunt (D, D, C)** |     |       |            |       |            |                   |          |       |
| Sparse-PFT             | 100 | -     | 5          | -     | -          | 50                |          |       |
| PFT-DPW                | 100 | -     | 2          | 1/10  | -          | 50                |          |       |
| POMCPOW                | 17  | -     | 6          | 1/100 | -          | 50                |          |       |
| AdaOPS                 | -   | -     | -          | -     | -          | 30                | 0.1      | 90    |
| POMCP                  | 17  | -     | -          | -     | -          | -                 |          |       |
| QMDP                   | -   | -     | -          | -     | -          | -                 |          |       |
| **VDP Tag (C, C, C)**  |     |       |            |       |            |                   |          |       |
| Sparse-PFT             | 70  | 20    | 8          | -     | -          | 10                |          |       |
| PFT-DPW                | 70  | 20    | 1/25       | 8     | 1/85       | 10                |          |       |
| POMCPOW                | 110 | 30    | 1/30       | 5     | 1/100      | 10                |          |       |
| **VDP Tag (C, D, C)**  |     |       |            |       |            |                   |          |       |
| Sparse-PFT             | 61  | -     | 10         | -     | -          | 10                |          |       |
| PFT-DPW                | 47  | -     | 4          | 1/5   | -          | 10                |          |       |
| POMCPOW                | 31  | -     | 5          | 1/20  | -          | 10                |          |       |
| AdaOPS                 | -   | -     | -          | -     | -          | 40                | 0.25     | 90    |

Table 2: Summary of hyperparameters used in experiments.

For Laser Tag, Light Dark, and Sub Hunt the leaf-node value estimation hyperparameters remain the same for all planners between experiments. PFT-DPW and Sparse-PFT use a partially observable rollout of a QMDP policy, POMCPOW uses a fully observable state value attained via offline state value iteration, and AdaOPS uses a random rollout for a lower bound estimate with QMDP value for an upper bound estimate. For both variants of Van Der Pol Tag, PFT-DPW and Sparse-PFT use the
mean value of random rollouts of all particle beliefs. Similarly, POMCPOW uses a single particle random rollout, and AdaOPS uses a random rollout for a lower bound estimate and $10^6$ as a static upper bound estimate.
References

Ayer, T., Alagoz, O., & Stout, N. K. (2012). A POMDP approach to personalize mammography screening decisions. *Operations Research, 60*(5), 1019–1034.

Bjarnason, R., Fern, A., & Tadepalli, P. (2009). Lower bounding Klondike solitaire with Monte-Carlo planning. *International Conference on Automated Planning and Scheduling (ICAPS).*

Bresina, J., Dearden, R., Meuleau, N., Ramakrishnan, S., Smith, D., Washington, R., Darwiche, A., & Friedman, N. (2002). Planning under continuous time and resource uncertainty: A challenge for AI. *Conference on Uncertainty in Artificial Intelligence (UAI).*

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., & Colton, S. (2012). A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games, 4*(1), 1–43.

Deglurkar, S., Lim, M. H., Tucker, J., Sunberg, Z. N., Faust, A., & Tomlin, C. J. (2021). Visual learning-based planning for continuous high-dimensional POMDPs [arXiv:2112.09456].

Egorov, M., Sunberg, Z. N., Balaban, E., Wheeler, T. A., Gupta, J. K., & Kochenderfer, M. J. (2017). POMDPs.jl: A framework for sequential decision making under uncertainty. *Journal of Machine Learning Research, 18*(26), 1–5.

Frew, E. W., Argrow, B., Borenstein, S., Swenson, S., Hirst, C. A., Havenga, H., & Houston, A. (2020). Field observation of tornadic supercells by multiple autonomous fixed-wing unmanned aircraft. *Journal of Field Robotics, 37*(6), 1077–1093.

Garg, N. P., Hsu, D., & Lee, W. S. (2019). DESPOT-alpha: Online POMDP planning with large state and observation spaces. *Robotics: Science and Systems.*

Hoerger, M., & Kurniawati, H. (2021). An on-line POMDP solver for continuous observation spaces. *Holland, J. E., Kochenderfer, M. J., & Olson, W. A. (2013). Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. Air Traffic Control Quarterly, 21*(3), 275–297.*

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence, 101*(1-2), 99–134.

Kearns, M., Mansour, Y., & Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning, 49*(2), 193–208.

Kochenderfer, M. J. (2015). *Decision making under uncertainty: Theory and application.* MIT Press.

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization.* MIT Press.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. *European Conference on Machine Learning (ECML), 282–293.*

Kurniawati, H., & Yadav, V. (2016). An online POMDP solver for uncertainty planning in dynamic environment. *Robotics research* (pp. 611–629). Springer.

Lim, M. H., Tomlin, C. J., & Sunberg, Z. N. (2020). Sparse tree search optimality guarantees in POMDPs with continuous observation spaces. *International Joint Conference on Artificial Intelligence (IJCAI).*

Lim, M. H., Tomlin, C. J., & Sunberg, Z. N. (2021). Voronoi progressive widening: Efficient online solvers for continuous state, action, and observation POMDPs. *IEEE Conference on Decision and Control (CDC).*

Littman, M. L., Cassandra, A. R., & Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. *International Conference on Machine Learning (ICML).*
Memarzadeh, M., & Boettiger, C. (2018). Adaptive management of ecological systems under partial observability. *Biological Conservation*, 224, 9–15.

Mern, J., Yildiz, A., Sunberg, Z., Mukerji, T., & Kochenderfer, M. J. (2021). Bayesian optimized Monte Carlo planning. *AAAI Conference on Artificial Intelligence (AAAI)*.

Metelli, A. M., Papini, M., Faccio, F., & Restelli, M. (2018). Policy optimization via importance sampling. *Advances in Neural Information Processing Systems (NIPS)*, 5442–5454.

Papadimitriou, C. H., & Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3), 441–450.

Ross, S., Pineau, J., Paquet, S., & Chaib-Draa, B. (2008). Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32, 663–704.

Seiler, K. M., Kurniawati, H., & Singh, S. P. N. (2015). An online and approximate solver for POMDPs with continuous action space. *IEEE International Conference on Robotics and Automation (ICRA)*, 2290–2297.

Shah, D., Xie, Q., & Xu, Z. (2022). Nonasymptotic analysis of Monte Carlo tree search. *Operations Research*, 0(0).

Shani, G., Pineau, J., & Kaplow, R. (2013). A survey of point-based POMDP solvers. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 27(1), 1–51.

Silver, D., & Veness, J. (2010). Monte-Carlo planning in large POMDPs. *Advances in Neural Information Processing Systems*, 2164–2172.

Singh, S. P., & Yee, R. C. (1994). An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3), 227–233.

Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5), 1071–1088.

Sunberg, Z., & Kochenderfer, M. J. (2018). Online algorithms for POMDPs with continuous state, action, and observation spaces. *International Conference on Automated Planning and Scheduling (ICAPS)*.

Sunberg, Z., & Kochenderfer, M. J. (2022). Improving automated driving through pomdp planning with human internal states. *IEEE Transactions on Intelligent Transportation Systems*, 1–11.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT Press.

Wu, C., Yang, G., Zhang, Z., Yu, Y., Li, D., Liu, W., & Hao, J. (2021). Adaptive online packing-guided search for POMDPs. *Advances in Neural Information Processing Systems (NeurIPS)*, 34.

Ye, N., Somani, A., Hsu, D., & Lee, W. S. (2017). DESPOT: Online POMDP planning with regularization. *Journal of Artificial Intelligence Research*, 58, 231–266.