Journal of Artificial Intelligence Research 77 (2023) 1591-1636 Submitted 12/2022; published 08/2023

Optimality Guarantees for Particle Belief Approximation of POMDPs

Michael H. Lim MICHAELHLIM@BERKELEY.EDU
University of California, Berkeley,
Electrical Engineering and Computer Sciences Department

Tyler J. Becker TYLER.BECKER-1@COLORADO.EDU
University of Colorado Boulder,
Aerospace Engineering Sciences Department

Mykel J. Kochenderfer MYKEL@STANFORD.EDU
Stanford University,
Aeronautics and Astronautics Department

Claire J. Tomlin TOMLIN@EECS.BERKELEY.EDU
University of California, Berkeley,
Electrical Engineering and Computer Sciences Department

Zachary N. Sunberg ZACHARY.SUNBERG@COLORADO.EDU
University of Colorado Boulder,
Aerospace Engineering Sciences Department

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 characterizing the approximation error of the particle filtering techniques that these algorithms use has not previously been proposed. Our main contribution is bounding the error between any POMDP and its corresponding finite sample particle belief MDP (PB-MDP) approximation. This fundamental bridge between PB-MDPs and POMDPs allows us to adapt any sampling-based MDP algorithm to a POMDP by solving the corresponding particle belief MDP, thereby extending the convergence guarantees of the MDP algorithm to the POMDP. Practically, this is implemented by using the particle filter belief transition model as the generative model for the MDP solver. While this requires access to the observation density model from the POMDP, it only increases the transition sampling complexity of the MDP solver by a factor of $\mathcal{O}(C)$, where C is the number of particles. Thus, when combined with sparse sampling MDP algorithms, this approach can yield algorithms for POMDPs that have no direct theoretical dependence on the size of the state and observation spaces. 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 problems with many local optima, because it only depends on a minimal set of problem structure requirements.

Recently proposed POMDP tree search algorithms (Garg et al., 2019; Hoerger & Kurniawati, 2021; 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; Hoerger & Kurniawati, 2021; Lim et al., 2021; Mern et al., 2021; Sunberg & Kochenderfer, 2018; Wu et al., 2021), there has been little progress on a general theory describing why this family of algorithms can enjoys such good performance. Though there have been some algorithm-specific results (outlined in Section 2.4), a considerable gap in the connection between POMDPs and practical approximations using particle methods still remains.

This manuscript formally justifies that optimality guarantees in a finite sample particle belief MDP (PB-MDP) approximation of a POMDP/belief MDP yield optimality guarantees in the original POMDP as well. We accomplish this 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-ω. Specifically, we prove that the Sparse Sampling-ω Q-value estimates are close to both optimal Q-values of the POMDP and PB-MDP with high probability. 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. This probability scales as $1-\mathcal{O}(C^D \exp(-t \cdot C))$, where C is the number of particles; D is the planning depth; and t is a number determined by the POMDP reward function and probability distributions, number of particles, and desired accuracy. Notably, this convergence rate does not directly depend on the size of the state space nor the observation space, but rather depends on the Rényi divergence that links the probabilities concerning state and observation trajectories and the planning horizon D.

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 to preserve the convergence guarantees in the POMDP. Practically, this means additionally assuming we have an explicit observation model $\mathcal{Z}$ and swapping out the state transition generative model with a particle filtering-based model. This change only increases the computational complexity of transition generation by a factor of $\mathcal{O}(C)$, 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, since 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).

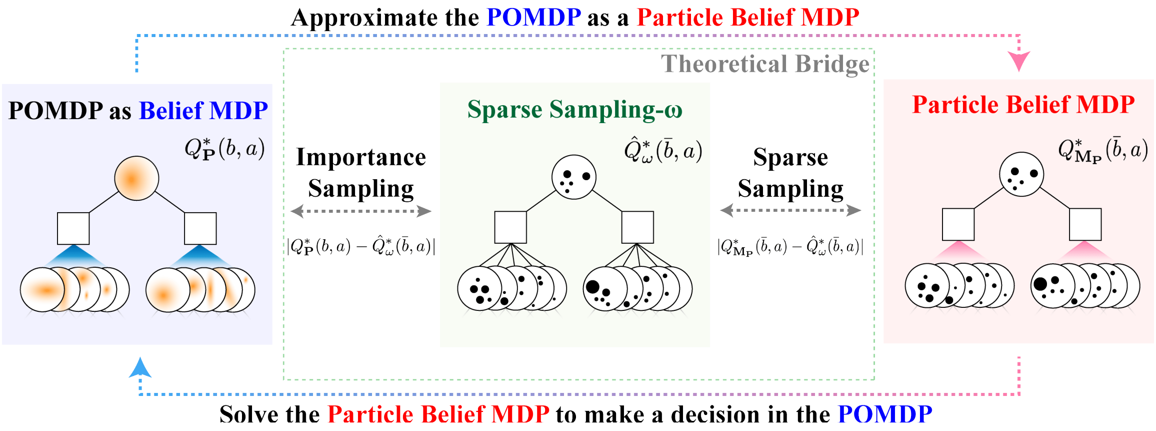

Approximate the POMDP as a Particle Belief MDP
Theoretical Bridge
POMDP as Belief MDP
$Q^*_{\mathbf{P}}(b,a)$
Importance Sampling
$|Q^*_{\mathbf{P}}(b,a) - \hat{Q}^*_{\omega}(\bar{b},a)|$
Sparse Sampling-ω
$\hat{Q}^*_{\omega}(\bar{b},a)$
Sparse Sampling
$|Q^*_{\mathbf{M_P}}(\bar{b},a) - \hat{Q}^*_{\omega}(\bar{b},a)|$
Particle Belief MDP
$Q^*_{\mathbf{M_P}}(\bar{b},a)$
Solve the Particle Belief MDP to make a decision in the POMDP

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.

Then, Section 4 introduces Sparse Sampling-ω 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 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 with a PB-MDP to solve a POMDP by applying an online Q-value-estimating algorithm repeatedly in a closed loop with observations from the environment. 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, and validates the improvements in performance of PB-MDP approximation with the increase in number of particles C while keeping other hyperparameters fixed.

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 $(S, A, O, \mathcal{T}, \mathcal{Z}, R, \gamma)$. 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 $\mathcal{T}$ and $\mathcal{Z}$ define state transitions and observation emissions, respectively. The transition probability distribution is conditioned on the current state s and action a and is denoted $\mathcal{T}(s' \mid s,a)$. The observation probability is conditioned on the previous action and current state[1] and denoted $\mathcal{Z}(o \mid a,s')$. 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 $\mathcal{T}$, $\mathcal{Z}$, or R. Such samples can be produced using a so-called *generative model* (Kearns et al., 2002) denoted with $s', o, r \leftarrow G(s,a)$. In some algorithms, only one or two of the outputs of G are used and the others are discarded, e.g. the notation "$o \leftarrow G(s,a)$" indicates that s' and r are discarded.

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

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)\right]. \tag{1}$$

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

$$h_t \equiv (b_0, a_0, o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t). \tag{2}$$

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,

$$b_t(s) \equiv \mathbb{P}(s_t = s \mid h_t). \tag{3}$$

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 both the immediate reward from taking the action in the current state and whether that action will lead to states favorable for attaining rewards in the future. For a history h and a corresponding belief b, the history-action and belief-action value functions, defined as

$$Q^\pi(h,a) \equiv Q^\pi(b,a) \equiv \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t) \,\middle|\, b_0 = b,\, a_0 = a,\, a_t = \pi(b_t)\right], \tag{4}$$

take both of these factors into account. When π 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 π is an optimal policy, these value functions are denoted with Q^* and V^*. If Q^* can be calculated, an optimal policy π^* can simply be extracted with $\pi^*(h) = \operatorname{argmax}_a Q^*(h,a)$. 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

[1] It is possible to use observation probability distributions additionally conditioned on the previous state, $\mathcal{Z}(o \mid s,a,s')$, in all algorithms discussed in this paper, but we use $\mathcal{Z}(o \mid a,s')$ for brevity.

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

$$\mathbb{E}_{X\sim\mathcal{P}}[f(X)] = \int f(x)\mathcal{P}(x)dx = \int f(x)\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}\mathcal{Q}(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\mathcal{P}(y_i)}{\mathcal{Q}(y_i)}f(y_i), \tag{5}$$

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, 2015; 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*, $\tilde{b} = \{s_i\}_{i=1}^N$. The density is approximated by $\tilde{b}(s) \approx \sum_{i=1}^N \delta(s_i = s)$, 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, $\tilde{b}' = \{s_i' : o_i = o\}$. 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, $\tilde{b} = \{(s_i, w_i)\}_{i=1}^N$. The density is approximated with $\tilde{b}(s) \approx \frac{\sum_{i=1}^N w_i \delta(s_i = s)}{\sum_{i=1}^N w_i}$. A belief update consists of simulating each particle once or more and then calculating the new weight according to the importance sampling correction: $w_i' = w_i \cdot Z(o \mid s, a, s_i')$. 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, 2015; Thrun et al., 2005).

2.3 Online POMDP Solvers

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)[2] (Silver & Veness, 2010) and constructs the tree by using Upper Confidence Bound (UCB), asymmetrically favoring regions of the history and action spaces that are likely to be visited when the optimal policy is executed (Bjarnason et al., 2009; Kocsis & Szepesvári, 2006; Shah et al., 2022).

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 unweighted particle belief representations, they will 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-α (Garg et al., 2019) incorporates a similar weighting scheme and uses the α-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.

2.4 Theoretical Analysis of Particle-based POMDP Algorithms

Several previous studies have analyzed particle-based POMDP algorithms from a theoretical perspective. Silver and Veness (2010) claim that POMCP value estimates converge to the optimal value for discrete POMDPs on the basis that it equivalent to applying UCT to the history MDP corresponding to the POMDP. However, as mentioned above, POMCP does not converge in continuous observation POMDPs (Lim et al., 2020; Sunberg & Kochenderfer, 2018). Ye et al. (2017) analyzed the approximation of a POMDP with a finite set of *scenarios*, which essentially correspond to random seeds that are fixed across different possible action sequences, and bounded the performance of the DESPOT algorithm that uses these scenarios. However, these bounds depend on the size of the observation space, $|O|$, and thus cannot be applied to continuous observation spaces. This analysis was expanded by Luo et al. (2019) to cover a case in which scenarios are selected from an importance distribution.

Bai et al. (2014) also provide convergence guarantees for their Monte Carlo value iteration (MCVI) algorithm which uses simulations in a manner somewhat akin to particle filtering. Their analysis extends to continuous observation spaces, but the algorithm is best suited for offline use,

[2] 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.

unlike the algorithms we focus on. According to Bai et al. (2014), MCVI spends hours computing a policy graph that can be executed quickly online.

Lim et al. (2020) presented the first theoretical analysis of online POMDP tree search algorithms that use weighted particle filtering. However, the partially observable weighted sparse sampling (POWSS) algorithm analyzed in that work is not efficient enough to be practically useful. Wu et al. (2021) provide analytical performance guarantees for a simplified version of AdaOPS, a recent particle belief tree search POMDP solver included in our numerical analysis in Section 6. However, the full AdaOPS algorithm used in the numerical experiments is more complex than the simplified version used in the theoretical portion of the work.

Du et al. (2021) analyzed the number of particles needed to control partially observable linear systems. Finally, there is a large body of work on particle filters without consideration of decision making. Some results from this field are summarized by Crisan and Doucet (2002).

In contrast to these works that provide guarantees for individual algorithms or limited cases, the analysis in this paper provides a general bound for particle belief approximation of a broad class of POMDPs, giving justification for MDP algorithms to be adapted to solve POMDPs 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* $\mathbf{P} = (S, A, O, \mathcal{T}, \mathcal{Z}, R, \gamma)$ *is the MDP* $\mathbf{M_P} = (\Sigma, A, \tau, \rho, \gamma)$ *defined by the following elements:*

- Σ: 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 $s_{d,i} \in S$, $w_{d,i} \in \mathbb{R}^+$. [3] For the sake of brevity in the rest of the paper, we drop $_{i=1}^{C}$ and render a particle belief as $\{s_{d,i}, w_{d,i}\}$. The beliefs are not assumed to be permutation invariant, meaning that particle beliefs with different particle orders are considered different elements in Σ. This simplifies derivation of the transition distribution (see Eq. (9)) because each particle transition is independent.

- A: Action space. Remains the same as the original action space.

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

$$w_{d+1,i} = w_{d,i} \cdot \mathcal{Z}(o \mid a, s_{d+1,i}). \tag{6}$$

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

$$\tau(\bar{b}_{d+1} \mid \bar{b}_d, a) \equiv \int_O \mathbb{P}(\bar{b}_{d+1} \mid \bar{b}_d, a, o) \mathbb{P}(o \mid \bar{b}_d, a) \, do. \tag{7}$$

The first term in the integrand product $\mathbb{P}(\bar{b}_{d+1} \mid \bar{b}_d, a, o)$ is the conditional transition density given some observation o. Since each new state particle is generated independently and the

[3] The d subscript, the number of steps, is included for subscript order consistency with the rest of the paper, but is not meaningful in this context.

likelihood weight updates are deterministic given $s_{d,i}, s_{d+1,i}, a$ and o, this term can be written in terms of $\mathcal{T}$ and $\mathcal{Z}$:

$$\begin{aligned}
\mathbb{P}(\bar{b}_{d+1} \mid \bar{b}_d, a, o) &= \mathbb{P}(\{s_{d+1,i}, w_{d+1,i}\} \mid \{s_{d,i}, w_{d,i}\}, a, o) && (8)\\
&= \prod_{i=1}^{C} \mathbb{P}(s_{d+1,i}, w_{d+1,i} \mid s_{d,i}, a, o) && (9)\\
&= \begin{cases} \prod_{i=1}^{C} \mathcal{T}(s_{d+1,i} \mid s_{d,i}, a) & \text{if } w_{d+1,i} = w_{d,i} \cdot \mathcal{Z}(o \mid a, s_{d+1,i}) \ \forall i \\ 0 & \text{otherwise.} \end{cases} && (10)
\end{aligned}$$

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

$$\begin{aligned}
\mathbb{P}(o \mid \bar{b}_d, a) = \mathbb{P}(o \mid \{s_{d,i}, w_{d,i}\}, a) &= \frac{\sum_{i=1}^{C} w_{d,i} \cdot \mathbb{P}(o \mid s_{d,i}, a)}{\sum_{i=1}^{C} w_{d,i}} && (11)\\
&= \frac{\sum_{i=1}^{C} w_{d,i} \cdot [\int_S \mathcal{Z}(o \mid a, s')\mathcal{T}(s' \mid s_{d,i}, a) ds']}{\sum_{i=1}^{C} w_{d,i}}. && (12)
\end{aligned}$$

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

- ρ: Reward function $\rho(\bar{b}_d, a)$:

$$\rho(\bar{b}_d, a) = \frac{\sum_i w_{d,i} \cdot R(s_{d,i}, a)}{\sum_i w_{d,i}}. \tag{13}$$

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

- γ: 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 algorithm 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 $Q^*_{\mathbf{P}}$ and PB-MDP $Q^*_{\mathbf{M_P}}$ are close with high probability.

4. Sparse Sampling-ω

In order to show that the optimal Q-values of the POMDP, $Q^*_{\mathbf{P}}$, and PB-MDP, $Q^*_{\mathbf{M_P}}$, are approximately equivalent, we first introduce an algorithm called Sparse Sampling-ω (sparse sampling with weights), which will serve as a theoretical bridge between POMDP and PB-MDP. Sparse Sampling-ω 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-ω takes inspiration from sparse sampling (Kearns et al., 2002) for continuous state MDPs, using particle belief

Algorithm 1 Sparse Sampling-ω

```
Global Variables: $\gamma, G, C, D$.

Procedure: GENPF($\bar{b}, a$)
Input: particle belief set $\bar{b} = \{(s_i, w_i)\}$, action $a$.
Output: New updated particle belief set $\bar{b}' = \{(s'_i, w'_i)\}$, mean reward $\rho$.
$s_o \leftarrow$ sample $s_i$ from $\bar{b}$ w.p. $\frac{w_i}{\sum_i w_i}$
$o \leftarrow G(s_o, a)$
for $i = 1, \ldots, C$ do
    $s'_i, r_i \leftarrow G(s_i, a)$
    $w'_i \leftarrow w_i \cdot \mathcal{Z}(o|a, s'_i)$
$\bar{b}' \leftarrow \{(s'_i, w'_i)\}_{i=1}^{C}$
$\rho \leftarrow \sum_i w_i r_i / \sum_i w_i$
return $\bar{b}', \rho$
```

```
Procedure: ESTIMATEV($\bar{b}, d$)
Input: particle belief set $\bar{b} = \{(s_i, w_i)\}$, depth $d$.
Output: A scalar $\hat{V}_d^*(\bar{b})$ that is an estimate of $V^*(\bar{b})$.
if $d \geq D$ then
    return 0
for $a \in A$ do
    $\hat{Q}_d^*(\bar{b}, a) \leftarrow$ ESTIMATEQ($\bar{b}, a, d$)
return $\hat{V}_d^*(\bar{b}) \leftarrow \max_{a \in A} \hat{Q}_d^*(\bar{b}, a)$

Procedure: ESTIMATEQ($\bar{b}, a, d$)
Input: particle belief set $\bar{b} = \{(s_i, w_i)\}$, action $a$, depth $d$.
Output: A scalar $\hat{Q}_d^*(\bar{b}, a)$ that is an estimate of $Q_d^*(b, a)$.
for $i = 1, \ldots, C$ do
    $\bar{b}'_i, \rho \leftarrow$ GENPF($\bar{b}, a$)
    $\hat{V}_{d+1}^*(\bar{b}'_i) \leftarrow$ ESTIMATEV($\bar{b}'_i, d+1$)
return $\hat{Q}_d^*(\bar{b}, a) \leftarrow \rho + \frac{1}{C}\sum_{i=1}^{C} \gamma \cdot \hat{V}_{d+1}^*(\bar{b}'_i)$
```

states. Note that Sparse Sampling-ω 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, this theoretically well-behaved algorithm is what lets us effectively bridge the gap between $Q^*_{\mathbf{P}}$ and $Q^*_{\mathbf{M_P}}$.

4.1 Algorithm Definition

The Sparse Sampling-ω algorithm is defined with the procedures listed in Algorithm 1. The global variables are the discount factor γ, the generative model G, the observation width and number of particles C, and the planning depth D. GENPF is the helper function to generate the next-step particle belief set, where the particles are evolved according to the transition density $\mathcal{T}$ and the weights are updated through the observation density $\mathcal{Z}$. In GENPF, the sampled states s'_i are inserted into each next-step particle belief set $\overline{bao_j}$ with the new weights $w'_i = w_i \cdot \mathcal{Z}(o_j \mid a, s'_i)$, which are the adjusted probability of hypothetically sampling observation o_j from state s'_i. Furthermore, the reward returned by 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 main planning functions in Sparse Sampling-ω are the ESTIMATEV and ESTIMATEQ procedures. 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 recursively calls ESTIMATEV to estimate the Q-function at a given step with a weighted average. In ESTIMATEQ, Sparse Sampling-ω samples the next particle belief state using GENPF.

Consequently, the Sparse Sampling-ω 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-ω is not computationally efficient as it fully expands the sparsely sampled tree with full particle belief states. It serves only to demonstrate theoretical convergence and is only practically applicable to very small toy POMDP problems.

Sparse Sampling-ω 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 algorithm (Lim et al., 2020). Specifically, whereas POWSS generates exactly one observation and corresponding new belief for each particle in a belief, Sparse Sampling-ω randomly selects a state to generate the observation each time GENPF is called in Line 2 of Algorithm 1. This means that Sparse Sampling-ω 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-ω 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-ω algorithm can be made to approximate both optimal Q-values of the POMDP $Q^*_{\mathbf{P}}$ and PB-MDP $Q^*_{\mathbf{M_P}}$ arbitrarily closely by increasing the observation width C. Theorem 2 proves that the Sparse Sampling-ω 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 $Q^*_{\mathbf{P}}$, and sparse sampling proof (Kearns et al., 2002) to prove the optimality in $Q^*_{\mathbf{M_P}}$.

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 $\mathcal{P}$ while the estimator only has access to another distribution $\mathcal{Q}$ along with the importance weights $w_{\mathcal{P}/\mathcal{Q}}(x) \propto \mathcal{P}(x)/\mathcal{Q}(x)$. This technique is crucial for Sparse Sampling-ω 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:

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

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

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

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

$$d_\infty(\mathcal{P}||\mathcal{Q}) = \operatorname*{ess\,sup}_{x\sim\mathcal{Q}} w_{\mathcal{P}/\mathcal{Q}}(x). \tag{14}$$

Assuming $d_\infty(\mathcal{P}||\mathcal{Q})$ is finite, we prove an estimator concentration bound in the following theorem.

Theorem 1 (SN d_∞-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 N independent identically distributed random variables with distribution $\mathcal{Q}$, 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 $1 - 3\exp(-N\cdot t^2(\lambda,N))$:*

$$|\mathbb{E}_{x\sim\mathcal{P}}[f(x)] - \tilde{\mu}_{\mathcal{P}/\mathcal{Q}}| \leq \lambda, \tag{15}$$

$$t(\lambda,N) \equiv \frac{\lambda}{\|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q})} - \frac{1}{\sqrt{N}}. \tag{16}$$

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_∞ 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_∞ 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-ω tree at all depths d reach convergence. The branching of the tree induces a factor proportional to C^D. Theorem 1 applied with $N = C$ will not only help offset the C^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-ω.

The following assumptions are needed for the Sparse Sampling-ω coupled convergence proof:

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

(ii) The densities $\mathcal{Z},\mathcal{T},b_0$ have the property that, for any observation sequence $\{o_{n,j}\}_{n=1}^{d}$, the Rényi divergence of the target distribution $\mathcal{P}^d$ and sampling distribution $\mathcal{Q}^d$ (Eqs. (22) and (23)) is bounded above by $d_\infty^{\max}$ for all $d = 0,\ldots,D-1$:

$$d_\infty(\mathcal{P}^d||\mathcal{Q}^d) = \text{ess sup}_{x\sim\mathcal{Q}^d} w_{\mathcal{P}^d/\mathcal{Q}^d}(x) \leq d_\infty^{\max} \tag{17}$$

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

(iv) We can sample from the generating function G and evaluate the observation density $\mathcal{Z}$.

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

(vi) We restrict our analysis to all the beliefs $b \in B$ that are realizable from the initial belief b_0 through Bayesian updates with action sequences $\{a_n\}$ and observation sequences $\{o_n\}$.

Intuitively, condition (ii) means that the ratio of the observation probability conditioned on the true state 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-ω.

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-ω 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 $\|f\|_\infty \leq f_{\max}$, and a particle belief state $\bar{b}_d = \{s_{d,i}, w_{d,i}\}$ at depth d represents b_d with particle likelihood weighting that is recursively updated as $w_{d,i} = w_{d-1,i} \cdot \mathcal{Z}(o_d \mid a, s_d)$. 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 the actions $\{a_n\}_{n=0}^{d-1}$ and observations $\{o_n\}_{n=1}^{d}$, for all beliefs $b_d \in B$ that are realizable given the initial belief b_0:*

$$\tilde{\mu}_{\bar{b}_d}[f] = \frac{\sum_{i=1}^{C} w_{d,i} f(s_{d,i})}{\sum_{i=1}^{C} w_{d,i}}, \tag{18}$$

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

$$|\mathbb{E}_{s \sim b_d}[f(s)] - \tilde{\mu}_{\bar{b}_d}[f]| \leq \lambda, \tag{19}$$

$$t_{\max}(\lambda, C) \equiv \frac{\lambda}{f_{\max} d_\infty^{\max}} - \frac{1}{\sqrt{C}}. \tag{20}$$

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 independent identically distributed random variable sequences of depth d, as GENPF independently generates each state sequence i according to the transition density $\mathcal{T}$. 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 is only through the particle likelihood weights.

We abbreviate some terms of interest with the following notation:

$$\mathcal{T}_{1:d}^{i} \equiv \prod_{n=1}^{d} \mathcal{T}(s_{n,i} \mid s_{n-1,i}, a_n); \quad \mathcal{Z}_{1:d}^{i} \equiv \prod_{n=1}^{d} \mathcal{Z}(o_n \mid a_n, s_{n,i}), \tag{21}$$

where d is the depth, and i is the index of the state sample. Intuitively, $\mathcal{T}^i_{1:d}$ is the transition density of the ith state sequence, $\{s_{n,i}\}_{n=1}^d$, and $\mathcal{Z}^i_{1:d}$ is the conditional density of observation sequence $\{o_n\}$ given the ith state sequence from the root node to depth d. Additionally, b^i_d denotes $b_d(s_{d,i})$ and $w_{d,i}$ the weight of $s_{d,i}$.

Then, we apply importance sampling to our system for all depths $d = 0,\ldots,D-1$. Here, $\mathcal{P}^d$ is the normalized measure of the state sequence $\{s_{n,i}\}_{n=0}^d$ conditioned on the observation sequence $\{o_n\}_{n=1}^d$ and action sequence $\{a_n\}_{n=0}^{d-1}$ up to the node at depth d, and $\mathcal{Q}^d$ is the measure of the state sequence conditioned only on the action sequence. For simplicity, we use $\mathcal{Z}_{1:d}$ to denote the product of observation likelihoods $\prod_{n=1}^d \mathcal{Z}(o_n \mid a_{n-1}, s_n)$ and $\mathcal{T}_{1:d}$ to denote the product of transition densities $\prod_{n=1}^d \mathcal{T}(s_n \mid s_{n-1}, a_{n-1})$. Then, for an arbitrary action sequence $\{a_n\}$, the following describes the densities necessary to define importance weighting:

$$\mathcal{P}^d = \mathcal{P}^d_{\{a_n,o_n\}}(\{s_{n,i}\}) = \frac{(\mathcal{Z}^i_{1:d})(\mathcal{T}^i_{1:d})b^i_0}{\int_{S^{d+1}}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0 ds_{0:d}} \tag{22}$$

$$\mathcal{Q}^d = \mathcal{Q}^d_{\{a_n\}}(\{s_{n,i}\}) = (\mathcal{T}^i_{1:d})b^i_0 \tag{23}$$

$$w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}) = \frac{(\mathcal{Z}^i_{1:d})}{\int_{S^{d+1}}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0 ds_{0:d}}. \tag{24}$$

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 can show that the recursive likelihood updating scheme in Lemma 1 produces valid likelihood weights up to a normalization by simply expanding the weight $w_{d,i}$:

$$w_{d,i} = w_{d-1,i} \cdot \mathcal{Z}(o_d \mid a_{d-1}, s_{d,i}) = w_{d-2,i} \prod_{n=d-1}^{d} \mathcal{Z}(o_n \mid a_{n-1}, s_{n,i}) = \ldots = \mathcal{Z}^i_{1:d} \propto w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}). \tag{25}$$

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

$$\tilde{\mu}_{\bar{b}_d}[f] = \frac{\sum_{i=1}^C w_{d,i} \cdot f(s_{d,i})}{\sum_{i=1}^C w_{d,i}} = \frac{\sum_{i=1}^C w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}) \cdot f(s_{d,i})}{\sum_{i=1}^C w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\})}. \tag{26}$$

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

Note that proving this lemma allows us to apply the particle likelihood weighting SN inequality whenever we encounter weighted averages with particle likelihood weights for a realizable particle belief. Also, this result does not depend on any specific choice of observation sequence $\{o_n\}$.

4.2.4 COUPLED CONVERGENCE OF SPARSE SAMPLING-ω.

The theorem below describes Sparse Sampling-ω's coupled convergence to both optimal Q-values of the POMDP $Q^*_{\mathbf{P}}$ and PB-MDP $Q^*_{\mathbf{M_P}}$, as C is increased.

Theorem 2 (Sparse Sampling-ω Coupled Optimality)**.** *Suppose conditions (i)-(vi) are satisfied. Then, for any $\lambda > 0$ and $0 < \delta \leq 1$, choosing particle count constant C that satisfies:*

$$C = \max\left\{ \left(\frac{4V_{\max} d_{\infty}^{\max}}{\lambda}\right)^2, \frac{64V_{\max}^2}{\lambda^2}\left(D \log \frac{24|A|^{\frac{D+1}{D}} V_{\max}^2 D}{\lambda^2} + \log\frac{1}{\delta}\right)\right\}, \tag{27}$$

the Q-function estimates $\hat{Q}^*_{\omega,d}(\bar{b}_d, a)$ obtained for all depths $d = 0, \ldots, D-1$, realized beliefs or histories b_d encountered in the Sparse Sampling-ω tree, and actions a are jointly near-optimal with respect to $Q^*_{\mathbf{P},d}$ and $Q^*_{\mathbf{M_P},d}$ with probability at least $1-\delta$:

$$|Q^*_{\mathbf{P},d}(b_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1-\gamma}, \tag{28}$$

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1-\gamma}. \tag{29}$$

To prove Theorem 2, we follow a similar proof strategy from our previous proof for POWSS (Lim et al., 2020) to show that Eq. (28) holds, and a similar strategy of the original sparse sampling proof (Kearns et al., 2002) to show that Eq. (29) 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 $Q^*_{\mathbf{P}}$ and PB-MDP $Q^*_{\mathbf{M_P}}$.

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 λ, δ. 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 $1 - 6|A|(4|A|C)^D \exp(-C \cdot \tilde{t}^2)$:*

$$|Q^*_{\mathbf{P},d}(b_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \alpha_d,\ \alpha_d = \lambda + \gamma\alpha_{d+1},\ \alpha_{D-1} = \lambda, \tag{30}$$

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \beta_d,\ \beta_d = \gamma(\lambda + \beta_{d+1}),\ \beta_{D-1} = 0, \tag{31}$$

$$t_{\max}(\lambda, C) = \frac{\lambda}{4V_{\max} d_{\infty}^{\max}} - \frac{1}{\sqrt{C}},\ \tilde{t} = \min\{t_{\max}, \lambda / 4\sqrt{2}V_{\max}\} \tag{32}$$

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 $Q^*_{\mathbf{P}}$ and the corresponding PB-MDP $Q^*_{\mathbf{M_P}}$ are

$$\begin{aligned}
Q^*_{\mathbf{P},d}(b_d, a) &= \mathbb{E}_{\mathbf{P}}[R(s_d, a) + \gamma V^*_{\mathbf{P},d+1}(b_d a o) \mid b_d] && (33)\\
&= \int_S R(s_d, a) b_d \cdot ds_d + \gamma \int_S \int_S \int_O V^*_{\mathbf{P},d+1}(b_d a o)(\mathcal{Z}_{d+1})(\mathcal{T}_{d,d+1}) b_d \cdot ds_{d:d+1} do, && (34)\\
Q^*_{\mathbf{M_P},d}(\bar{b}_d, a) &= \rho(\bar{b}, a) + \gamma \mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1}) \mid \bar{b}_d, a] && (35)\\
&= \frac{\sum_i w_{d,i} \cdot R(s_{d,i}, a)}{\sum_{i=1}^C w_{d,i}} + \gamma \int_\Sigma V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1}) \tau(\bar{b}_{d+1} \mid \bar{b}_d, a) d\bar{b}_{d+1}. && (36)
\end{aligned}$$

The Sparse Sampling-ω value estimates are mathematically equal to

$$\hat{V}^*_{\omega,d}(\bar{b}_d) = \max_{a \in A} \hat{Q}^*_{\omega,d}\left(\bar{b}_d, a\right) \tag{37}$$

$$\hat{Q}^*_{\omega,d}(\bar{b}_d, a) = \frac{\sum_{i=1}^{C} w_{d,i} r_{d,i}}{\sum_{i=1}^{C} w_{d,i}} + \frac{1}{C}\sum_{i=1}^{C} \gamma \cdot \hat{V}^*_{\omega,d+1}\left(\bar{b}'^{[I_i]}_{d+1}\right), \tag{38}$$

where $\{I_i\}$ are C independent identically distributed random variables with finite discrete distribution $p_{w,d}$ with probability mass $p_{w,d}(I = j) = (w_{d,j} / \sum_k w_{d,k})$, and particle belief state $\bar{b}'^{[I_i]}_{d+1}$ is updated by an observation generated from s_{d,I_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. (30) is satisfied, which is an adapted and substantially 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):

$$\begin{aligned} |Q^*_{\mathbf{P},d}(b_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq & \underbrace{\left| \mathbb{E}_{\mathbf{P}}[R(s_d, a) \mid b_d] - \frac{\sum_{i=1}^{C} w_{d,i} r_{d,i}}{\sum_{i=1}^{C} w_{d,i}} \right|}_{\text{(A)}} \\ & + \gamma \underbrace{\left| \mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_d a o) \mid b_d] - \frac{1}{C}\sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1}) \right|}_{\text{(B)}} \end{aligned} \tag{39}$$

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. This also proves the base case.

To bound the next-step value estimation error (B), we introduce particle likelihood SN estimators and Monte Carlo average estimators to bridge the following quantities (detailed definitions and bounds of each terms are in Appendix D):

$$\begin{aligned} & \underbrace{\left| \mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_d a o) \mid b_d] - \frac{1}{C}\sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1}) \right|}_{\text{(B)}} \\ & \leq \underbrace{\left| \mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_d a o) \mid b_d] - \frac{\sum_{i=1}^{C} w_{d,i} \mathbf{V}^*_{\mathbf{P},d+1}(b_d, a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} \right|}_{\text{(1) Importance sampling error}} + \underbrace{\left| \frac{\sum_{i=1}^{C} w_{d,i} \mathbf{V}^*_{\mathbf{P},d+1}(b_d, a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} - \frac{1}{C}\sum_{i=1}^{C} \mathbf{V}^*_{\mathbf{P},d+1}(b_d, a)^{[I_i]} \right|}_{\text{(2) MC weighted sum approximation error}} \\ & + \underbrace{\left| \frac{1}{C}\sum_{i=1}^{C} \mathbf{V}^*_{\mathbf{P},d+1}(b_d, a)^{[I_i]} - \frac{1}{C}\sum_{i=1}^{C} V^*_{\mathbf{P},d+1}(b_d a o^{[I_i]}) \right|}_{\text{(3) MC next-step integral approximation error}} + \underbrace{\left| \frac{1}{C}\sum_{i=1}^{C} V^*_{\mathbf{P},d+1}(b_d a o^{[I_i]}) - \frac{1}{C}\sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1}) \right|}_{\text{(4) Inductive function estimation error}}. \end{aligned} \tag{40}$$

PB-MDP Value Convergence: Second, we show that Eq. (31) is satisfied, which is an adapted and substantially modified proof of sparse sampling convergence (Kearns et al., 2002). Once again, we split the difference between the SN estimator and the $Q^*_{\mathbf{M_P}}$ function into two terms, the reward estimation error (A) and the next-step value estimation error (B):

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \underbrace{\left|\rho(\bar{b}_d, a) - \rho(\bar{b}_d, a)\right|}_{\text{(A) = 0}} + \gamma \underbrace{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1}) \mid \bar{b}_d, a] - \frac{1}{C}\sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(B)}} \tag{41}$$

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 inductive function estimation error (detailed definitions and bounds of each terms are in Appendix D):

$$\underbrace{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1}) \mid \bar{b}_d, a] - \frac{1}{C}\sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(B)}} \tag{42}$$

$$\leq \underbrace{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1}) \mid \bar{b}_d, a] - \frac{1}{C}\sum_{i=1}^{C} V^*_{\mathbf{M_P},d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(1) MC transition approximation error}} + \underbrace{\left|\frac{1}{C}\sum_{i=1}^{C} V^*_{\mathbf{M_P},d+1}(\bar{b}'^{[I_i]}_{d+1}) - \frac{1}{C}\sum_{i=1}^{C} \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(2) Inductive function estimation error}}.$$

Combining the probability bounds used in both of these procedures results in a worst case $\mathcal{O}(\exp(-t \cdot C))$ 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 $\mathcal{O}(C^D)$. Therefore, we can obtain the Q-value estimator concentration inequality, with convergence rate $\mathcal{O}(C^D \exp(-t \cdot C))$. □

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 $\mathbf{M_P}$. Theorem 3 shows that the Q-values $Q^*_{\mathbf{P}}$ and $Q^*_{\mathbf{M_P}}$ are close to each other with high probability, and Theorem 4 shows that using any approximately optimal MDP planning algorithm $\mathcal{A}$ in the particle belief MDP $\mathbf{M_P}$ as a policy yields near-optimal value in the original POMDP if applied repeatedly in a closed loop with the environment and an exact belief updater.

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 $\mathbf{M_P}$. In essence, this theorem claims that the two optimal Q-values $Q^*_{\mathbf{P}}$ and $Q^*_{\mathbf{M_P}}$ are close with high probability, because creating a very accurate Q-value estimator via Sparse Sampling-ω that is close to both $Q^*_{\mathbf{P}}$ and $Q^*_{\mathbf{M_P}}$ happens with high probability.

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

$$|Q^*_{\mathbf{P}}(b,a) - Q^*_{\mathbf{M_P}}(\bar{b},a)| \leq \varepsilon_{\mathbf{M_P}}. \tag{43}$$

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

$$\begin{aligned}
|Q^*_{\mathbf{P}}(b,a) - Q^*_{\mathbf{M_P}}(\bar{b},a)| &\leq |Q^*_{\mathbf{P}}(b,a) - \hat{Q}^*_{\omega}(\bar{b},a)| + |\hat{Q}^*_{\omega}(\bar{b},a) - Q^*_{\mathbf{M_P}}(\bar{b},a)| && (44)\\
&\equiv |Q^*_{\mathbf{P},0}(b_0,a) - \hat{Q}^*_{\omega,0}(\bar{b}_0,a)| + |Q^*_{\mathbf{M_P},0}(\bar{b}_0,a) - \hat{Q}^*_{\omega,0}(\bar{b}_0,a)| && (45)\\
&\leq \frac{2\lambda}{1-\gamma} \equiv \varepsilon_{\mathbf{M_P}}. && (46)
\end{aligned}$$

Since this bound holds with high probability for creating any hypothetical Sparse Sampling-ω tree, this must mean that $|Q^*_{\mathbf{P}}(b,a) - Q^*_{\mathbf{M_P}}(\bar{b},a)| \leq \varepsilon_{\mathbf{M_P}}$ in general with high probability.

The convergence rate of $\delta_{\mathbf{M_P}}$ is $\mathcal{O}(C^D \exp(-\tilde{t}\cdot C))$. This means that as we increase the number of particles, we can expect better performance by approximately solving a POMDP via particle belief 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 $\mathcal{A}$, if the optimal planner can approximate Q-values with arbitrary precision $\varepsilon_{\mathcal{A}}$ with probability at least $1-\delta_{\mathcal{A}}$ in the corresponding particle belief MDP of a given POMDP, then the planning algorithm can approximate the POMDP Q-values within $\varepsilon_{\mathbf{M_P}} + \varepsilon_{\mathcal{A}}$ with probability at least $1-\delta_{\mathbf{M_P}}-\delta_{\mathcal{A}}$:*

$$|Q^*_{\mathbf{P}}(b,a) - \hat{Q}^{\mathcal{A}}_{\mathbf{M_P}}(\bar{b},a)| \leq \varepsilon_{\mathbf{M_P}} + \varepsilon_{\mathcal{A}}. \tag{47}$$

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

Note that it would also be possible to devise an expected value version of the bounds by converting the probability statement into an expected value statement.

Essentially, Corollary 1 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

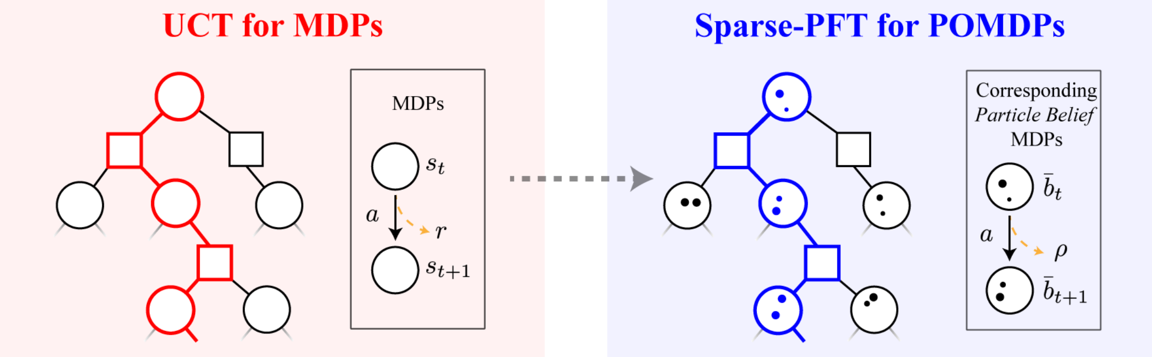


Figure 2: Illustration of promoting an MDP algorithm (UCT) into a POMDP algorithm (Sparse-PFT). This practically only involves changing the transition generative model into a particle filtering-based transition generative model that deals with particle belief states.

similar optimality guarantees. The most remarkable thing about this result is that it does not directly depend on the size of the state space nor the observation space. However, the dependence may indirectly come through the observation density and thus the Rényi divergence factor, and in practice, the generative model sampling complexity often depends on the dimensionality of the state space. Moreover, even though this approach is insensitive to the state and observation space size, the guarantees and practical algorithms are highly sensitive to the planning horizon D.

In most practical cases, this method would usually incur an additional $\mathcal{O}(C)$ compute time factor in a given transition sampling step as single particle belief state generation now needs to propagate C particles forward instead of a single particle/state. Moreover, if the algorithm requires storing the beliefs, the memory requirements are increased by an $\mathcal{O}(C)$ factor compared with the MDP algorithm. Fortunately, as demonstrated in Section 6, a modest number of particles often gives adequate performance in practice.

Proving Corollary 1 allows us to prove Theorem 4 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. In this context, we mean a near-optimal MDP planning algorithm $\mathcal{A}$ to be an algorithm with small values of $\varepsilon_{\mathcal{A}}, \delta_{\mathcal{A}}$ that would satisfy the conditions required in the proof of the theorem. Examples of such algorithms include sparse sampling (Kearns et al., 2002) and others (Bjarnason et al., 2009; Couëtoux et al., 2011). The detailed proof for Theorem 4 is in Appendix E.

Theorem 4 (Particle Belief MDP Approximate Policy Convergence)**.** *Suppose a near-optimal MDP planning algorithm $\mathcal{A}$ is used to plan with particle belief MDP $\mathbf{M_P}$ repeatedly in a closed loop with POMDP environment $\mathbf{P}$ and an exact Bayesian belief updater to process observations from the environment. Further assume that regularity conditions (i)-(vi) are met for $\mathbf{M_P}$ and that $\mathcal{A}$ can approximate the Q-values of $\mathbf{M_P}$ with arbitrary precision $\varepsilon_{\mathcal{A}}$ with probability at least $1-\delta_{\mathcal{A}}$. Then, for any $\varepsilon > 0$, we can choose C such that the value obtained by planning with $\mathcal{A}$ in $\mathbf{M_P}$ is within ε of the optimal POMDP value function at b_0:*

$$V^*_{\mathbf{P}}(b_0) - V^{\mathcal{A}}_{\mathbf{M_P}}(b_0) \le \varepsilon. \tag{48}$$

Algorithm 2 Sparse-PFT Algorithm

```
Global Variables: $\gamma, n, c_{\text{UCB}}, \beta_{\text{UCB}}, G, D$.

Procedure: PLAN($b$)
Input: Belief $b$.
Output: An action $a$.
 1: for $i = 1, \ldots, C$ do
 2:     $s \leftarrow$ sample from $b$
 3:     $\bar{b} \leftarrow \bar{b} \cup \{(s, 1/C)\}$
 4: for $i = 1, \ldots, n$ do
 5:     SIMULATE($\bar{b}, 0$)
 6: return $a \leftarrow \arg\max_{a \in C(b)} Q(b, a)$

Procedure: 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: $a \leftarrow \arg\max_{a \in C(\bar{b})} Q(\bar{b}, a) + c_{\text{UCB}} \frac{N(\bar{b})^{\beta_{\text{UCB}}}}{\sqrt{N(\bar{b}, a)}}$
 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 $q$
```

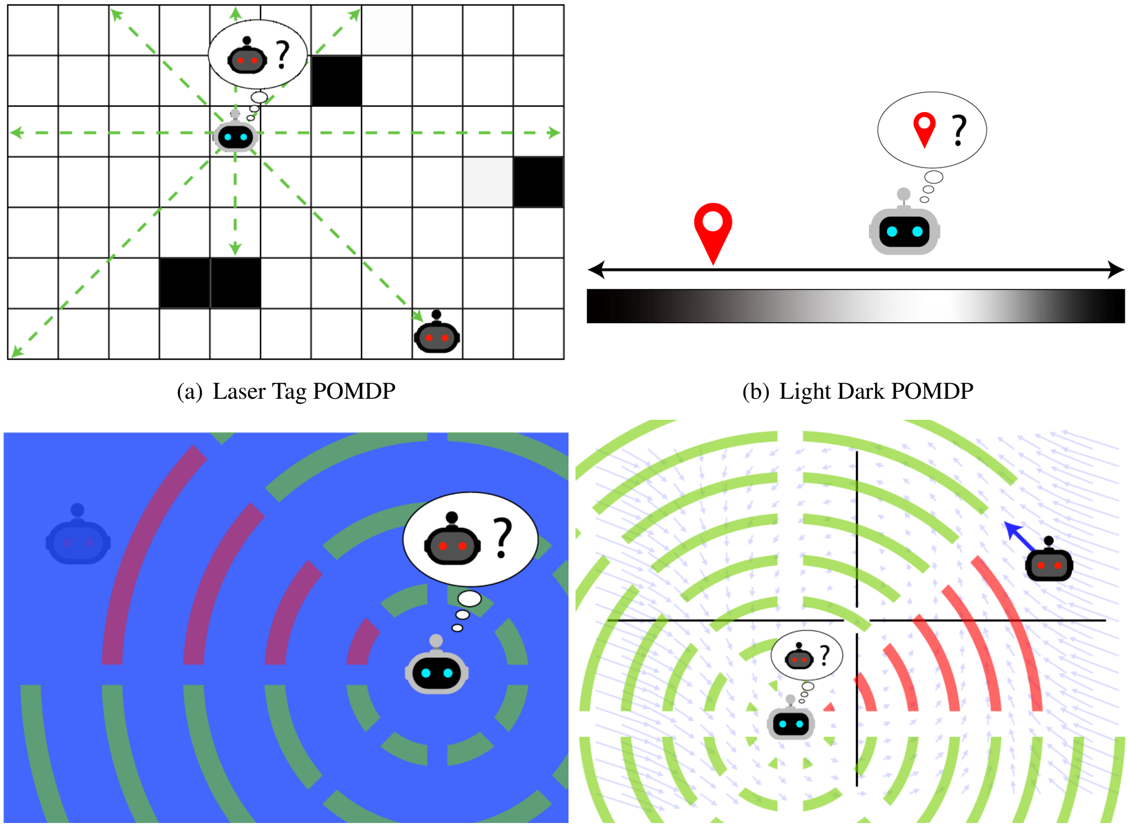
5.3 Sparse Particle Filter Tree (Sparse-PFT)

By utilizing the results in Theorem 3 and Theorem 4, we can promote a variant of sampling-based MDP planning algorithm Upper Confidence Tree (UCT), Sparse UCT (Bjarnason et al., 2009), 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 an algorithm that is simple to implement, and enjoys both theoretical guarantees and high performance in practice.

The entry point of Sparse-PFT is the PLAN procedure which repeatedly calls the the SIMULATE procedure to construct the tree and choose an action. Both of these procedures are defined in Algorithm 2. The set of global variables for Sparse-PFT includes the same global variables used for Sparse Sampling-ω with the addition of n, the number of tree search queries, and c_{UCB} and β_{UCB}, the polynomial Upper Confidence Bound parameters that determine the amount of exploration in Line 3 (Shah et al., 2022). 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 that Sparse-PFT manages particle belief sets through GENPF (defined in Algorithm 1) 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



(a) Laser Tag POMDP

(b) Light Dark POMDP

(c) Sub Hunt POMDP

(d) VDP Tag POMDP

Figure 3: Illustration of the four environments we test our algorithms on: (a) Laser Tag, (b) Light Dark, (c) Sub Hunt, and (d) VDP Tag (discrete and continuous versions).

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.

The following sections contain descriptions of the evaluation problems along with discussion of solver performance. 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 closed-loop planning, whenever an observation was received from the environment, the belief was updated with a particle filter independent of the particle filter used in planning, and no part of the planning tree was saved for re-use on subsequent steps as done by Silver and Veness (2010). Since the observations received from the environment in this outer simulation loop were not generated from state particles in the filter, a larger number particles compared to GenPF are used to ensure likely states are present. 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. We also vary belief particle count C with Simulate calls held constant to demonstrate the effect of particle belief approximation resolution on the quality of the resulting policy in Fig. 10 by using the optimized hyperparameters from Table 2 with 1000 simulation experiments for Laser Tag, Light Dark and Sub Hunt, and 100 for VDP Tag and Discrete VDP Tag. 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.

6.1 Laser Tag

The Laser Tag POMDP (Fig. 5) 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 $\mathcal{N}(d, 2.5)$ 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.

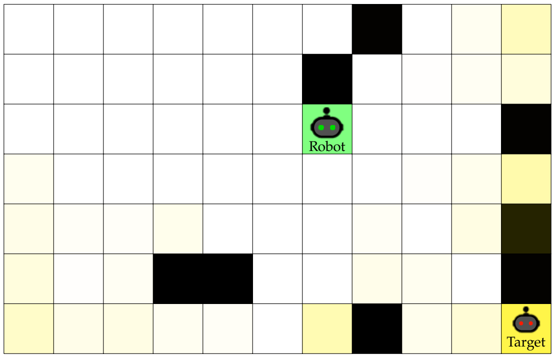


Figure 5: Example Sparse-PFT operating in Laser Tag: as indicated by the belief (yellow), the agent (green) has roughly located the evading robot at bottom right corner.

From the results, we find that the PFT methods outperform both POMCPOW and AdaOPS: PFT-DPW consistently outperforms both planners across different planning times, and Sparse-PFT outperforms all other planners with increased planning time. This suggests that for Laser Tag, having a full particle belief approximation rather than dynamically varying particle size is helpful for keeping track of likely particle hypotheses. Furthermore, this also suggests that the double progressive widening has diminishing returns when the action space has a fixed small size.

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 (Sunberg & Kochenderfer, 2018). This hinders POMCP's ability to develop a non-myopic multi-step plan and yield accurate

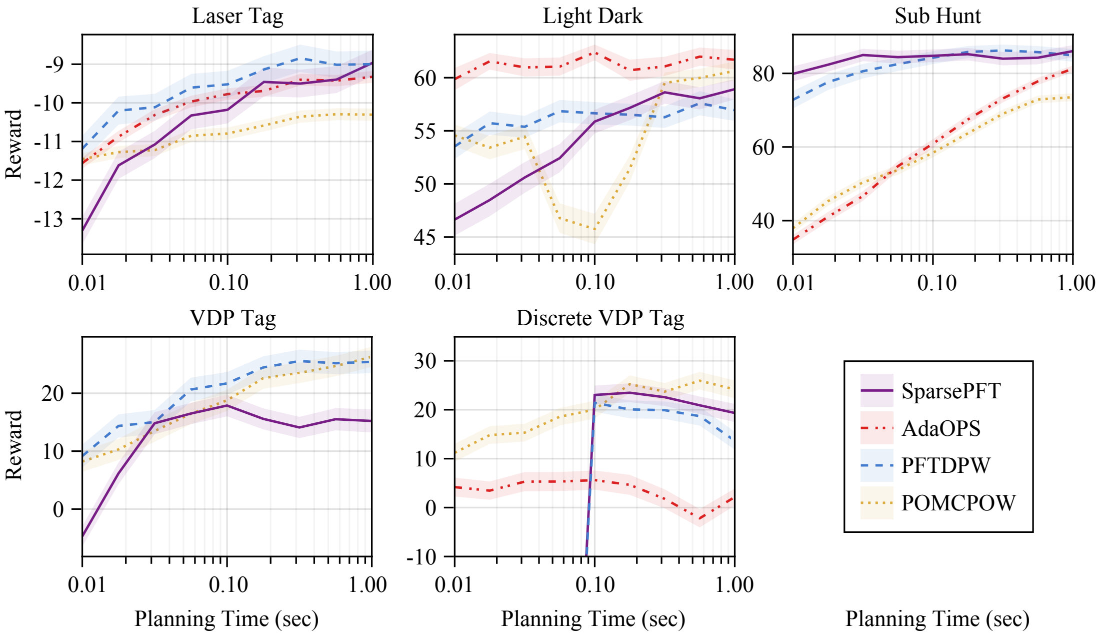


Figure 4: Comparative benchmark results summary over different planning time allotments. Ribbons surrounding the Monte Carlo benchmark mean estimates indicate two standard error confidence bands. Planning times are given in log scale.

	Laser Tag (D, D, D)		Light Dark (D, D, C)		Sub Hunt (D, D, C)	
Sparse-PFT	$-\mathbf{8.96 \pm 0.17}$	🟩	58.9 ± 0.5	🟩	$\mathbf{86.0 \pm 0.8}$	🟩
PFT-DPW	$-\mathbf{8.99 \pm 0.07}$	🟩	56.9 ± 0.5	🟩	84.9 ± 0.9	🟩
POMCPOW	-10.3 ± 0.08	🟩	60.6 ± 0.4	🟩	73.5 ± 0.7	🟩
AdaOPS	$\dagger -\mathbf{9.33 \pm 0.08}$	🟩	$\mathbf{61.7 \pm 0.5}$	🟩	81.3 ± 0.6	🟩
QMDP	-10.4 ± 0.08	🟩	3.28 ± 0.5	🟧	28.0 ± 0.6	🟥
POMCP	-16.0 ± 0.09	🟥	-14.86 ± 2.3	🟥	30.0 ± 0.8	🟥
Random Policy	-51.0 ± 0.18		-85.0 ± 0.72		4.20 ± 0.27	
	VDP Tag (C, C, C)		VDP Tag^D (C, D, C)			
Sparse-PFT	15.2 ± 1.0	🟩	19.3 ± 0.9	🟩		
PFT-DPW	$\mathbf{25.4 \pm 1.0}$	🟩	13.7 ± 0.9	🟩		
POMCPOW	$\mathbf{26.3 \pm 0.9}$	🟩	$\mathbf{24.2 \pm 0.9}$	🟩		
AdaOPS			2.0 ± 0.4	🟥		
Random Policy	-66.8 ± 0.24		-66.6 ± 0.25			

Table 1: Tabular comparative benchmark summary. All results are given as (mean ± standard error) for a 1 second planning time allotment. The algorithm(s) with the best average performance within one standard error is shown in boldface for each experiment. The three letters after each problem name indicate whether the state, action, and observation spaces are continuous or discrete, respectively. † For AdaOPS, the result obtained by Wu et al. (2021) on Laser Tag was -8.31 ± 0.18, so this entry was also bolded.

action values, empirically showing the importance of particle weighting. On the other hand, QMDP exhibits performance similar to the modern solvers. While the agent does not initially know its own location, it has sufficient information to localize using the laser sensor observations after some steps, and the evading robot behavior leads it reliably to the corners. Thus, since this problem requires less active information gathering, the crude QMDP approximation performs well.

6.2 Light Dark

The 1-dimensional Light Dark POMDP is designed to require 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,

$$R(s,a) = \begin{cases} +100 & \text{if } s=0, a=0 \\ -100 & \text{if } s\neq 0, a=0 \\ -1 & \text{otherwise} \end{cases}, \tag{49}$$

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 $s=10$. The observation distribution is $\mathcal{N}(s, |s-10|+\varepsilon)$, where ε 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.

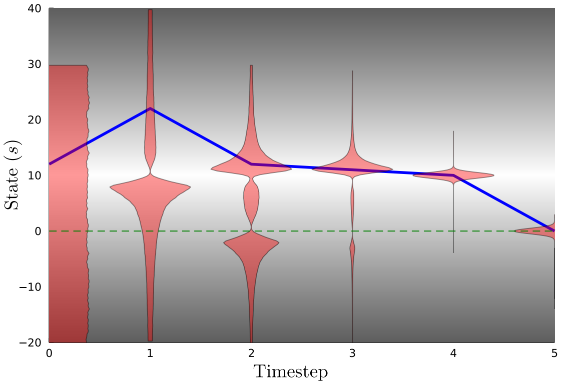


Figure 6: Example Sparse-PFT trajectory for Light Dark, successfully localizing in the light region to reach the goal in the dark region.

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 region via certainty-equivalent control, and then takes action $a=-10$ down to the goal. This heuristic policy yields an expected reward of 62.0 ± 0.19 which is as good or better than any planner shown in Table 1.

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 when the PFT planner is supplied with a 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 Light Dark requires costly information gathering, QMDP performs suboptimally. Specifically, regardless of belief distribution entropy, QMDP myopically steers directly towards the goal location but rarely commits to taking action 0 within the simulation horizon due to high state uncertainty. POMCP also performs poorly due to the high branching factor introduced by the continuous observation space (Sunberg & Kochenderfer, 2018).

6.3 Sub Hunt

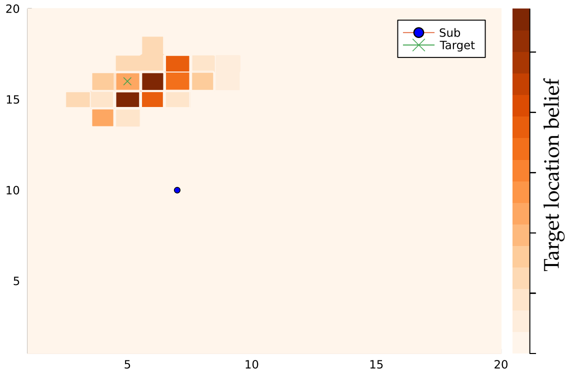


Figure 7: Example Sparse-PFT policy behavior for Sub Hunt, where the agent's submarine pursues the target with active information gathering.

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, with Sparse-PFT having a slight edge over all other planners. Because the state space is discrete, value iteration can be used to calculate Q-values for the fully observable MDP, and QMDP 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. POMCP and QMDP perform poorly due to the large branching factor (Sunberg & Kochenderfer, 2018) and inability to perform costly information gathering, respectively.

6.4 VDP Tag

The Van Der Pol Tag (VDP 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,

$$\dot{x} = \mu\left(x - \frac{x^3}{3} - y\right), \quad \dot{y} = \frac{x}{\mu}, \tag{50}$$

for which we use scaling constant $\mu = 2$. Because this problem has a continuous state space ($\mathcal{S} = \mathbb{R}^4$), a continuous action space ($\mathcal{A} = [0,2\pi) \times \{0,1\}$) and a continuous observation space ($\mathcal{O} = \mathbb{R}^8$), 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. For the action-

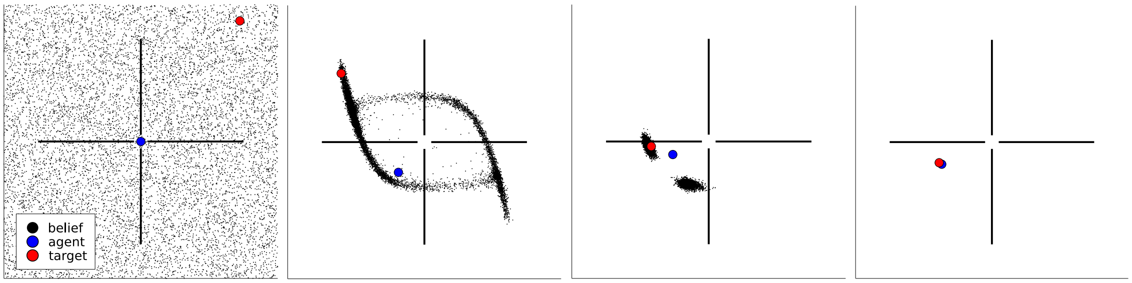


Figure 8: Example Sparse-PFT policy behavior for VDP Tag, which shows a successful localization of the target following the Van Der Pol dynamics, then a successful capture.

discretized VDP Tag, the available movement directions are 20 evenly spaced angles from 0° to 360°.

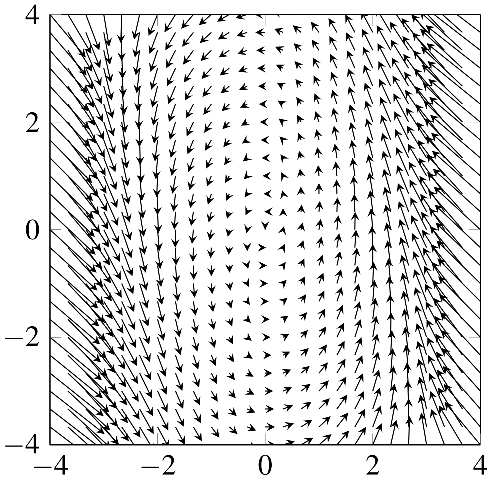


Figure 9: An example Van Der Pol vector field ($\mu = 0.5$) which defines the target dynamics. Unlike the agent, the target is not blocked by the barriers.

The continuous VDP 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. Furthermore, PFT-DPW and POMCPOW have similar performances across different planning times, suggesting that the main challenge of VDP Tag is being able to handle continuity of state, action, and observation spaces, while the particle belief approximation resolution does not affect the performance as much.

For discrete VDP Tag, 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 ($|\mathcal{A}| = 20$) 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 a computationally expensive simulator on all particles takes a long time, leading to very shallow trees.

6.5 Experimental Validation of Particle Belief Approximation Convergence

In order to test the effect of particle belief approximation resolution, or the number of belief particles C, on planner performance, we vary C while fixing the number of SIMULATE calls and using optimal hyperparameters found in Appendix F for Sparse-PFT planner. By increasing the the num-

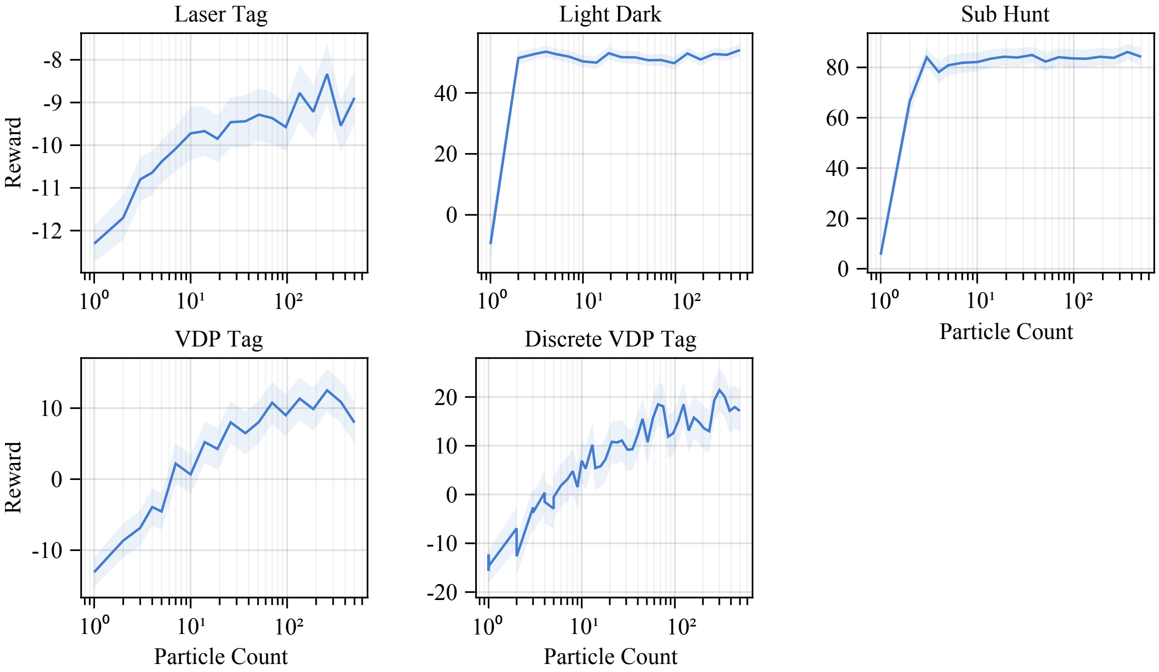


Figure 10: Sparse-PFT policy mean rewards with two standard error confidence band, varying belief particle count while holding number of tree search SIMULATE queries constant.

ber of belief particles, the particle belief becomes a more accurate representation of the actual belief function, and should lead to a better optimal Q-value estimation.

Across all five problem domains, increasing the number of particles C results in a roughly monotonic non-decreasing performance gain as shown in Fig. 10. In particular, we see a gradual performance increase for Laser Tag, VDP Tag, and Discrete VDP Tag as we increase the number of particles, while Light Dark and Sub Hunt problems reach their performance capacity rather quickly at less than 10 particles. However, when applying this principle to promote MDP algorithms into PB-MDP algorithms, the particle filtering transition generative model requires an extra $\mathcal{O}(C)$ computation and memory factor, so it is important to balance computational resource needs and value estimation accuracy when deploying these algorithms in practice.

These results offer two valuable insights. First, the experiment outcomes are consistent with the general trend suggested by the theoretical analysis: increasing the number of particles results in improved performance, presumably because the Q-value estimates are more likely to be accurate as established in Section 5. Second, not all problems benefit the same way from increasing the number of particles. Light Dark and Sub Hunt have rather simple state spaces, and adding more particles did not significantly improve the policy performance. In contrast, the other three problems continually benefited from having increased resolution of belief approximation. The belief approximation resolution is not the only factor contributing to the problem difficulty, but also other factors like action space cardinality and existence of simple rollout policies that perform well will contribute to the problem difficulty.

7. Conclusion

In this work, we formally show that optimality guarantees in a finite sample particle belief MDP (PB-MDP) approximation of a POMDP 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-ω 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. 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 to preserve the convergence guarantees in the POMDP. The transformation only increases the computational complexity of transition generation by a factor of $\mathcal{O}(C)$ by using particle filtering-based generative models. Our convergence result is not directly dependent on the size of the state space nor the observation space, but rather dependent on the Rényi divergence that links the probabilities concerning state and observation trajectories. This motivates particle belief-based POMDP algorithms such as Sparse Particle Filter Tree (Sparse-PFT), which enjoys algorithmic simplicity, theoretical guarantees, and practicality.

There are many interesting avenues for future research. First, the broader theoretical justification of more complex algorithms, such as POMCPOW and DESPOT-α, still do not exist. Showing theoretical validity of these algorithms would help to 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. Also, the particle number sweep suggests a method to characterize the difficulty of a POMDP problem, which may be of interest for both practitioners and researchers alike. 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, NSF National Robotics Initiative Grant No. 2133141, and the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1752814 and DGE 2146752. 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_∞-Concentration Bound

Theorem 1 (SN d_∞-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 N independent identically distributed random variables with distribution $\mathcal{Q}$, 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* $1-3\exp(-N\cdot t^2(\lambda,N))$*:*

$$|\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}|\leq\lambda, \tag{1}$$

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

Proof. This proof follows similar proof steps as in Metelli *et al.* (Metelli et al., 2018). Since we have upper bounds on the infinite 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 $\|g\|_\infty=d_\infty(\mathcal{P}||\mathcal{Q})\|f\|_\infty$:

$$\begin{aligned}
\mathbb{P}\left(\hat{\mu}_{\mathcal{P}/\mathcal{Q}}-\mathbb{E}_{x\sim P}[f(x)]\geq\lambda\right)&=\mathbb{P}\left(\hat{\mu}_{\mathcal{P}/\mathcal{Q}}-\mathbb{E}_{x\sim\mathcal{Q}}[\hat{\mu}_{\mathcal{P}/\mathcal{Q}}(x)f(x)]\geq\lambda\right) &(3)\\
&\leq\exp\left(-\frac{2N^2\lambda^2}{\sum_{i=1}^N 2(d_\infty(\mathcal{P}||\mathcal{Q})\|f\|_\infty)^2}\right) &(4)\\
&\leq\exp\left(-\frac{N\lambda^2}{d_\infty^2(\mathcal{P}||\mathcal{Q})\|f\|_\infty^2}\right)\equiv\delta &(5)\\
\mathbb{P}\left(|\hat{\mu}_{\mathcal{P}/\mathcal{Q}}-\mathbb{E}_{x\sim P}[f(x)]|\geq\lambda\right)&\leq 2\exp\left(-\frac{N\lambda^2}{d_\infty^2(\mathcal{P}||\mathcal{Q})\|f\|_\infty^2}\right)=2\delta &(6)
\end{aligned}$$

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:

$$\begin{aligned}
&\mathbb{P}(|\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}|\geq\lambda) &(7)\\
&\quad\leq\mathbb{P}(\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}-\mathbb{E}_{x\sim\mathcal{P}}[f(x)]\geq\lambda)+\mathbb{P}(\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}\geq\lambda) &(8)\\
&\quad\leq\mathbb{P}(\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}-\mathbb{E}_{x\sim\mathcal{Q}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}]\geq\tilde{\lambda})+\mathbb{P}(\mathbb{E}_{x\sim\mathcal{Q}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}\geq\tilde{\lambda}) &(9)\\
&\quad\leq\tilde{\delta}+\mathbb{P}(\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}\geq\lambda) &(10)
\end{aligned}$$

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

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

$$\tilde{\delta}=\exp\left(-\frac{N\tilde{\lambda}^2}{d_\infty^2(\mathcal{P}||\mathcal{Q})\|f\|_\infty^2}\right) \tag{12}$$

Note that the bias term in the SN estimator is bounded by following through Cauchy-Schwarz inequality, closely following steps from Metelli et al. (2018):

$$|\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\mathbb{E}_{x\sim\mathcal{Q}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}]| = \left|\mathbb{E}_{x\sim\mathcal{Q}}[\hat{\mu}_{\mathcal{P}/\mathcal{Q}}-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}]\right| \le \mathbb{E}_{x\sim\mathcal{Q}}[|\hat{\mu}_{\mathcal{P}/\mathcal{Q}}-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}|] \tag{13}$$

$$\le \mathbb{E}_{x\sim\mathcal{Q}}\left|\frac{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)f(x_i)}{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)}-\frac{1}{N}\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)f(x_i)\right| \tag{14}$$

$$= \mathbb{E}_{x\sim\mathcal{Q}}\left[\left|\frac{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)f(x_i)}{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)}\right|\left|1-\frac{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)}{N}\right|\right] \tag{15}$$

$$\le \mathbb{E}_{x\sim\mathcal{Q}}\left[\left(\frac{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)f(x_i)}{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)}\right)^2\right]^{1/2}\mathbb{E}_{x\sim\mathcal{Q}}\left[\left(1-\frac{\sum_{i=1}^N w_{\mathcal{P}/\mathcal{Q}}(x_i)}{N}\right)^2\right]^{1/2} \tag{16}$$

$$\le \|f\|_\infty\sqrt{\frac{d_2(\mathcal{P}||\mathcal{Q})-1}{N}} \le \|f\|_\infty\frac{d_\infty(\mathcal{P}||\mathcal{Q})}{\sqrt{N}} \tag{17}$$

In the last step, the first term is bounded by $\|f\|_\infty$ as the function is bounded, and the second term is bounded by the fact that we can bound the square root of variance with the supremum squared, where we square it for the convenience of the definition of $t(\lambda,N)$ 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 $\lambda > \|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q})/\sqrt{N}$ to bound the $\tilde{\delta}$ term:

$$\tilde{\delta} \le \exp\left(-\frac{N(\lambda-\|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q})/\sqrt{N})^2}{d_\infty^2(\mathcal{P}||\mathcal{Q})\|f\|_\infty^2}\right) \tag{18}$$

$$= \exp\left(-N\left(\frac{\lambda-\|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q})/\sqrt{N}}{\|f\|_\infty d_\infty(\mathcal{P}||\mathcal{Q})}\right)^2\right) \tag{19}$$

$$\equiv \exp\left(-N\cdot t^2(\lambda,N)\right) \tag{20}$$

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

$$\mathbb{P}(\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}\ge\lambda)\le\mathbb{P}(\mathbb{E}_{x\sim\mathcal{Q}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}\ge\tilde{\lambda}) \tag{21}$$

$$\le\mathbb{P}(|\mathbb{E}_{x\sim\mathcal{Q}}[\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}|\ge\tilde{\lambda})\le 2\tilde{\delta} \tag{22}$$

Thus, we obtain the following bound:

$$\mathbb{P}(|\mathbb{E}_{x\sim\mathcal{P}}[f(x)]-\tilde{\mu}_{\mathcal{P}/\mathcal{Q}}|\ge\lambda)\le 3\exp(-N\cdot t^2(\lambda,N)) \tag{23}$$

□

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

In the main paper, we show that $\tilde{\mu}_{\bar{b}_d}[f]$ 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 $\|f\|_\infty \leq f_{\max}$, and a particle belief state $\bar{b}_d = \{(s_{d,i}, w_{d,i})\}_{i=1}^{C}$ at depth d represents b_d with particle likelihood weighting that is recursively updated as $w_{d,i} = w_{d-1,i} \cdot \mathcal{Z}(o_d \mid a, s_d)$. 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 the actions $\{a_n\}_{n=0}^{d-1}$ and observations $\{o_n\}_{n=1}^{d}$, for all beliefs $b_d \in B$ that are realizable given the initial belief b_0:*

$$\tilde{\mu}_{\bar{b}_d}[f] = \frac{\sum_{i=1}^{C} w_{d,i} f(s_{d,i})}{\sum_{i=1}^{C} w_{d,i}}, \tag{24}$$

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

$$|\mathbb{E}_{s \sim b_d}[f(s)] - \tilde{\mu}_{\bar{b}_d}[f]| \leq \lambda, \tag{25}$$

$$t_{\max}(\lambda, C) \equiv \frac{\lambda}{f_{\max} d_\infty^{\max}} - \frac{1}{\sqrt{C}}. \tag{26}$$

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 $\mathcal{T}$.

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

$$\mathcal{T}_{1:d}^{i} \equiv \prod_{n=1}^{d} \mathcal{T}(s_{n,i} \mid s_{n-1,i}, a_n); \;\; \mathcal{Z}_{1:d}^{i} \equiv \prod_{n=1}^{d} \mathcal{Z}(o_n \mid a_n, s_{n,i}). \tag{27}$$

Here d denotes the depth, i denotes the index of the state sample. Intuitively, $\mathcal{T}_{1:d}^{i}$ is the transition density of state sequence i from the root node to depth d, and $\mathcal{Z}_{1:d}^{i}$ is the conditional density of observation sequence 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 $s_{d,i}$.

First, we show that $\tilde{\mu}_{\bar{b}_d}[f]$ 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:

$$b_{D-1}(s_{D-1}) = \frac{\int_{S^{D-1}} (\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-2}}{\int_{S^D} (\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-1}} \tag{28}$$

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

$$\mathbb{E}_{s \sim b_d}[f(s)] = \int_S f(s_{D-1}) b_{D-1} ds_{D-1} = \frac{\int_{S^D} f(s_{D-1})(\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-1}}{\int_{S^D} (\mathcal{Z}_{1:D-1})(\mathcal{T}_{1:D-1}) b_0 ds_{0:D-1}} \tag{29}$$

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 $\{o_n\}$. Normally, these generated observation sequences through GENPF will be correlated. For now, we treat the observation sequences $\{o_n\}$ as 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 the observation sequence conditioned on the state sequence i and action sequence 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 $\{o_n\}$.

$$\mathcal{P}^d = \mathcal{P}^d_{\{a_n,o_n\}}(\{s_{n,i}\}) = \frac{(\mathcal{Z}^i_{1:d})(\mathcal{T}^i_{1:d})b^i_0}{\int_{S^{d+1}}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0 ds_{0:d}} \tag{30}$$

$$\mathcal{Q}^d = \mathcal{Q}^d_{\{a_n\}}(\{s_{n,i}\}) = (\mathcal{T}^i_{1:d})b^i_0 \tag{31}$$

$$w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}) = \frac{(\mathcal{Z}^i_{1:d})}{\int_{S^{d+1}}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0 ds_{0:d}} \tag{32}$$

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:

$$w_{d,i} = w_{d-1,i} \cdot \mathcal{Z}(o_d \mid a_d, s_{d,i}) \propto \mathcal{Z}^i_{1:d}. \tag{33}$$

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

$$\tilde{\mu}_{\bar{b}_d}[f] = \frac{\sum_{i=1}^C (\mathcal{Z}^i_{1:d}) f(s_{d,i})}{\sum_{i=1}^C (\mathcal{Z}^i_{1:d})} = \frac{\sum_{i=1}^C \frac{(\mathcal{Z}^i_{1:d})}{\int_{S^D}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0 ds_{0:d}} f(s_{d,i})}{\sum_{i=1}^C \frac{(\mathcal{Z}^i_{1:d})}{\int_{S^D}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0 ds_{0:d}}} \tag{34}$$

$$= \frac{\sum_{i=1}^C w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}) f(s_{d,i})}{\sum_{i=1}^C w_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\})} = \sum_{i=1}^C \tilde{w}_{\mathcal{P}^d/\mathcal{Q}^d}(\{s_{n,i}\}) f(s_{d,i}) \tag{35}$$

Since $\{s_n\}_1, \ldots, \{s_n\}_C$ are independent identically distributed random variable 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 $d_\infty^{\max}$ a.s., we can bound the resulting $t_d(\lambda, C)$ by $t_{\max}(\lambda, C)$ a.s.:

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

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 $\{o_n\}$, where $\{o_n\}$ could be a sequence of random variables correlated with any elements of $\{s_{n,i}\}$. Thus, for any $\{o_n\}$,

$$|\mathbb{E}_{s\sim b_d}[f(s)] - \tilde{\mu}_{\bar{b}_d}[f]| \leq \lambda \tag{37}$$

holds with probability at least $1 - 3\exp(-C \cdot t^2_{\max}(\lambda, C))$. □

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 $d = 0,\dots,D-1$ and a, the following bounds hold with probability at least* $1-6|A|(4|A|C)^D\exp(-C\cdot\tilde{t}^2)$*:*

$$|Q^*_{\mathbf{P},d}(b_d,a)-\hat{Q}^*_{\omega,d}(\bar{b}_d,a)| \leq \alpha_d,\ \alpha_d = \lambda+\gamma\alpha_{d+1},\ \alpha_{D-1}=\lambda, \tag{38}$$

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d,a)-\hat{Q}^*_{\omega,d}(\bar{b}_d,a)| \leq \beta_d,\ \beta_d = \gamma(\lambda+\beta_{d+1}),\ \beta_{D-1}=0, \tag{39}$$

$$t_{\max}(\lambda,C) = \frac{\lambda}{4V_{\max}d_\infty^{\max}} - \frac{1}{\sqrt{C}},\ \tilde{t} = \min\{t_{\max}, \lambda/4\sqrt{2}V_{\max}\} \tag{40}$$

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 λ.

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

$$\begin{aligned}|Q^*_{\mathbf{P},d}(b_d,a)-\hat{Q}^*_{\omega,d}(\bar{b}_d,a)| \leq{}& \underbrace{\left|\mathbb{E}_{\mathbf{P}}[R(s_d,a)\mid b_d] - \frac{\sum_{i=1}^C w_{d,i}r_{d,i}}{\sum_{i=1}^C w_{d,i}}\right|}_{\text{(A)}} \\ &+\gamma\underbrace{\left|\mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_d ao)\mid b_d] - \frac{1}{C}\sum_{i=1}^C \hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(B)}}\end{aligned} \tag{41}$$

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 $\bar{b}'^{[I_i]}_{d+1}$ is updated by an observation generated from s_{d,I_i}. This reflects the fact that GENPF randomly selects a state particle s_o with probability $w_{d,o}/\sum_j w_{d,j}$ C times independently to generate a new observation for the next step particle belief state. Similarly, a particle belief state $\bar{b}'^{[i]}_{d+1}$ 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:

$$|Q^*_{\mathbf{P},D-1}(b_{D-1},a)-\hat{Q}^*_{\omega,D-1}(\bar{b}_{D-1},a)| \leq \underbrace{\left|\mathbb{E}_{\mathbf{P}}[R(s_{D-1},a)\mid b_{D-1}] - \frac{\sum_{i=1}^C w_{D-1,i}r_{D-1,i}}{\sum_{i=1}^C w_{D-1,i}}\right|}_{\text{(A)}}. \tag{42}$$

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 λ 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 λ terms when we add (A) and (B). This also covers the base case since $\alpha_{D-1} = \lambda \geq \frac{R_{\max}}{4V_{\max}}\lambda$.

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 inductive function estimation error bounded by α_{d+1}:

$$\underbrace{\left|\mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_dao) \mid b_d] - \frac{1}{C}\sum_{i=1}^{C}\hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(B)}} \tag{43}$$

$$\leq \underbrace{\left|\mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_dao) \mid b_d] - \frac{\sum_{i=1}^{C} w_{d,i}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}}\right|}_{\text{(1) Importance sampling error}} + \underbrace{\left|\frac{\sum_{i=1}^{C} w_{d,i}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} - \frac{1}{C}\sum_{i=1}^{C}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]}\right|}_{\text{(2) MC weighted sum approximation error}}$$

$$+ \underbrace{\left|\frac{1}{C}\sum_{i=1}^{C}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]} - \frac{1}{C}\sum_{i=1}^{C}V^*_{\mathbf{P},d+1}(b_dao^{[I_i]})\right|}_{\text{(3) MC next-step integral approximation error}} + \underbrace{\left|\frac{1}{C}\sum_{i=1}^{C}V^*_{\mathbf{P},d+1}(b_dao^{[I_i]}) - \frac{1}{C}\sum_{i=1}^{C}\hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(4) Inductive function estimation error}}.$$

$$\leq \underbrace{\frac{1}{4}\lambda}_{(1)} + \underbrace{\frac{1}{4}\lambda}_{(2)} + \underbrace{\frac{1}{2}\lambda}_{(3)} + \underbrace{\alpha_{d+1}}_{(4)}. \tag{44}$$

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 $\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a,s_{d,i}) \equiv \mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[i]}$ 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 $\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a,s_d)$:

$$\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[i]} \equiv \int_S\int_O V^*_{\mathbf{P},d+1}(b_dao)\mathcal{Z}(o \mid a,s_{d+1})\mathcal{T}(s_{d+1} \mid s_{d,i},a)ds_{d+1}do \tag{45}$$

$$\mathbb{E}_{\mathbf{P}}[V^*_{\mathbf{P},d+1}(b_dao) \mid b_d] = \int_S\int_S\int_O V^*_{\mathbf{P},d+1}(b_dao)(\mathcal{Z}_{d+1})(\mathcal{T}_{d,d+1})b_d \cdot ds_{d:d+1}do \tag{46}$$

$$= \int_S \mathbf{V}^*_{\mathbf{P},d+1}(b_d,a,s_d)b_d \cdot ds_d \tag{47}$$

$$= \frac{\int_{S^{d+1}}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a,s_d)(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0ds_{0:d}}{\int_{S^{d+1}}(\mathcal{Z}_{1:d})(\mathcal{T}_{1:d})b_0ds_{0:d}}. \tag{48}$$

Noting that the term (1) is then the difference between the SN estimator and the conditional expectation, and that $\left\|\mathbf{V}^*_{\mathbf{P},d+1}\right\|_\infty \leq V_{\max}$, 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 $t_{\max}$, the bound holds with probability at least $1 - 3\exp(-C \cdot t^2_{\max}(\lambda, C))$.

(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 bound. First, we assume that all the variables except I are given, which are $\{s_{d,i}, w_{d,i}\}, b_d, a$. Then, we note that $\mathbf{V}^*_{\mathbf{P},d+1}(b_d, a, \cdot)$ is a function bounded by $V_{\max}$. For convenience of notation and conceptual clarity, we will denote $\mathbf{V}^*_{\mathbf{P},d+1}(b_d, a)^{[i]} \equiv \mathbf{V}(i)$, which means the value estimate realization for the i-th state index. Noting that the probability mass is $p_{w,d}(I = i) = (w_{d,i}/\sum_j w_{d,j})$, the Monte Carlo summation error can be simplified as the following:

$$\left|\frac{\sum_{i=1}^{C} w_{d,i}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} - \frac{1}{C}\sum_{i=1}^{C}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]}\right| \Longrightarrow \left|\sum_{i=1}^{C} p_{w,d}(I=i)\cdot\mathbf{V}(i) - \frac{1}{C}\sum_{i=1}^{C}\mathbf{V}(I_i)\right|. \tag{49}$$

The first term in the difference is the expectation of $\mathbf{V}(\cdot)$ under the probability measure $p_{w,d}$:

$$\left|\sum_{i=1}^{C} p_{w,d}(I=i)\cdot\mathbf{V}(i) - \frac{1}{C}\sum_{i=1}^{C}\mathbf{V}(I_i)\right| = \left|\mathbb{E}_{p_{w,d}}[\mathbf{V}(I)] - \frac{1}{C}\sum_{i=1}^{C}\mathbf{V}(I_i)\right|. \tag{50}$$

This is precisely the form of the double-sided Hoeffding-type bound on the function values $\mathbf{V}(I)$, where a Monte Carlo summation, or the Monte Carlo average in this case, attempts to approximate the expected value. Therefore, we can choose λ such that the absolute difference is bounded by λ with probability at least $1 - 2\exp(-C\lambda^2/2V_{\max}^2)$ for an arbitrary fixed set of $\{s_{d,i}, w_{d,i}\}, b_d, a$:

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

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 $\mathbf{V}(I_i)$ 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:

$$\begin{aligned}
\mathbb{P}\left\{\Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \leq \lambda\right\} &= \mathbb{E}[\mathbf{1}_{\{\Delta(\{s_{d,i},w_{d,i}\},b_d,a,\{I_i\})\leq\lambda\}}] && (52)\\
&= \mathbb{E}\left[\mathbb{E}\left[\mathbf{1}_{\{\Delta(\{s_{d,i},w_{d,i}\},b_d,a,\{I_i\})\leq\lambda\}} \mid \{s_{d,i}, w_{d,i}\}, b_d, a\right]\right] && (53)\\
&= \mathbb{E}\left[\mathbb{P}\left\{\Delta(\{s_{d,i}, w_{d,i}\}, b_d, a, \{I_i\}) \leq \lambda \mid \{s_{d,i}, w_{d,i}\}, b_d, a\right\}\right] && (54)\\
&\geq \mathbb{E}[1 - 2\exp(-C\lambda^2/2V_{\max}^2)] && (55)\\
&= 1 - 2\exp(-C\lambda^2/2V_{\max}^2). && (56)
\end{aligned}$$

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 $1-2\exp(-C\lambda^2/32V_{\max}^2)$, which gets us our desired result:

$$\left|\frac{\sum_{i=1}^{C} w_{d,i}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[i]}}{\sum_{i=1}^{C} w_{d,i}} - \frac{1}{C}\sum_{i=1}^{C}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]}\right| \le \frac{\lambda}{4}. \tag{57}$$

(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 $\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]}$, we can simply use the quantity $V^*_{\mathbf{P},d+1}(b_d ao^{[I_i]})$, as the random vector (s_{d+1,I_i}, o_{I_i}) is jointly generated using G according to the correct probability $\mathcal{Z}(o \mid a, s_{d+1})\mathcal{T}(s_{d+1} \mid s_{d,I_i}, a)$ given s_{d,I_i} in the simulation realized in the tree. Consequently, the quantity $V^*_{\mathbf{P},d+1}(b_d ao^{[I_i]})$ for a given (s_{d,I_i}, b_d, a) is an unbiased 1-sample MC estimate of $\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]}$. We define the difference between these two quantities as Δ_{d+1}, which is implicitly a function of random variables (s_{d+1,I_i}, o_{I_i}):

$$\Delta_{d+1}(b_d,a)^{[I_i]} \equiv \mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]} - V^*_{\mathbf{P},d+1}(b_d ao^{[I_i]}). \tag{58}$$

Then, we note that $\|\Delta_{d+1}\|_\infty \le 2V_{\max}$ and $\mathbb{E}\Delta_{d+1} = 0$ by the Tower property conditioning on (s_{d,I_i}, b_d, a) (which is implicitly conditioning on I_i, but this does not matter greatly as everything cancels out) and integrating over (s_{d+1,I_i}, o_{I_i}) first, which holds for any choice of well-behaved sampling distributions on $\{s_{0:d}\}_i$. 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 $\|\Delta_{d+1}\|_\infty \le 2V_{\max}$, our λ factor is then augmented by 1/2 to once again obtain probability at least $1-2\exp(-C\lambda^2/32V_{\max}^2)$:

$$\left|\frac{1}{C}\sum_{i=1}^{C}\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]} - \frac{1}{C}\sum_{i=1}^{C}V^*_{\mathbf{P},d+1}(b_d ao^{[I_i]})\right| \tag{59}$$

$$= \left|\frac{1}{C}\sum_{i=1}^{C}(\mathbf{V}^*_{\mathbf{P},d+1}(b_d,a)^{[I_i]} - V^*_{\mathbf{P},d+1}(b_d ao^{[I_i]})) - 0\right|$$

$$= \left|\frac{1}{C}\sum_{i=1}^{C}\Delta_{d+1}(b_d,a)^{[I_i]} - \mathbb{E}\Delta_{d+1}\right| \le \frac{\lambda}{2}. \tag{60}$$

(4) **Inductive 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 α_{d+1}.

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

$$|Q^*_{\mathbf{P},d}(b_d,a) - \hat{Q}^*_d(\bar{b}_d,a)| \le \frac{R_{\max}}{4V_{\max}}\lambda + \gamma\left[\frac{1}{4}\lambda + \frac{1}{4}\lambda + \frac{1}{2}\lambda + \alpha_{d+1}\right] \tag{61}$$

$$\le \frac{1-\gamma}{4}\lambda + \gamma\left[\frac{1}{4}\lambda + \frac{3}{4\gamma}\lambda + \alpha_{d+1}\right] \tag{62}$$

$$= \lambda + \gamma\alpha_{d+1} = \alpha_d. \tag{63}$$

Now, we derive the worst case union bound probability. First, we want to ensure that the SN concentration inequality holds with probability $1-3\exp(-C\cdot t_{\max}^2(\lambda,C))$ 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 $1-2\exp(-C\lambda^2/32V_{\max}^2)$ 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

$$\max(3\exp(-C\cdot t_{\max}^2(\lambda,C)),2\exp(-C\lambda^2/32V_{\max}^2)) \tag{64}$$

$$\le 3\exp(-C\cdot t_{\max}^2(\lambda,C))+2\exp(-C\lambda^2/32V_{\max}^2) \tag{65}$$

$$\le 5\exp(-C\cdot \tilde{t}^2). \tag{66}$$

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 $d=0,\dots,D-1$, 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 $5|A|(4|A|C)^D\exp(-C\cdot\tilde{t}^2)$. 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 $1-5|A|(4|A|C)^D\exp(-C\cdot\tilde{t}^2)$ for all levels d:

$$|Q^*_{\mathbf{P},d}(b_d,a)-\tilde{Q}^*_d(\bar{b}_d,a)| \le \tilde{\alpha}_d. \tag{67}$$

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

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d,a)-\hat{Q}^*_{\omega,d}(\bar{b}_d,a)| \le \underbrace{\left|\rho(\bar{b}_d,a)-\rho(\bar{b}_d,a)\right|}_{\text{(A)}=0}+\gamma\underbrace{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1})\mid\bar{b}_d,a]-\frac{1}{C}\sum_{i=1}^{C}\hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(B)}}. \tag{68}$$

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 $d=0,\dots,D-1$. We use the triangle inequality repeatedly to separate it into two terms; (1) the MC transition approximation error bounded by λ, and (2) the inductive function estimation error bounded by β_{d+1}:

$$\underbrace{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1})\mid\bar{b}_d,a]-\frac{1}{C}\sum_{i=1}^{C}\hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(B)}} \tag{69}$$

$$\le \underbrace{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1})\mid\bar{b}_d,a]-\frac{1}{C}\sum_{i=1}^{C}V^*_{\mathbf{M_P},d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(1) MC transition approximation error}}+\underbrace{\left|\frac{1}{C}\sum_{i=1}^{C}V^*_{\mathbf{M_P},d+1}(\bar{b}'^{[I_i]}_{d+1})-\frac{1}{C}\sum_{i=1}^{C}\hat{V}^*_{\omega,d+1}(\bar{b}'^{[I_i]}_{d+1})\right|}_{\text{(2) Inductive function estimation error}}$$

$$\le \underbrace{\lambda}_{(1)}+\underbrace{\beta_{d+1}}_{(2)}. \tag{70}$$

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 $\{\bar{b}'^{[I_i]}_{d+1}\}$ given $(\bar{b}_d, a)$ is essentially approximating the integration over the transition density $\tau(\bar{b}_{d+1} \mid \bar{b}_d, a)$. Since the value function and its estimate are both bounded by $V_{\max}$, we can invoke Hoeffding bound here to obtain the following exponential probabilistic bound on the difference:

$$\mathbb{P}\left\{\left|\mathbb{E}_{\mathbf{M_P}}[V^*_{\mathbf{M_P},d+1}(\bar{b}_{d+1}) \mid \bar{b}_d, a] - \frac{1}{C}\sum_{i=1}^{C} V^*_{\mathbf{M_P},d+1}(\bar{b}'^{[I_i]}_{d+1})\right| \leq \lambda\right\} \geq 1 - 2\exp(-C\lambda^2/2V^2_{\max}). \tag{71}$$

(2) **Inductive Function Estimation 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 β_{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 $1 - |A|(|A|C)^D \exp(-C\lambda^2/2V^2_{\max})$. Thus, since (A) is 0 and (B) is bounded by $\lambda + \beta_{d+1}$, the Q-value estimation error with respect to $\mathbf{M_P}$ is bounded as desired:

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \gamma(\lambda + \beta_{d+1}) = \beta_d. \tag{72}$$

Combining both concentration bounds: In order to enable simultaneous satisfaction of the two concentration inequalities, we bound the worst case union probability by using the definition of $\tilde{t}$ and combining the upper bounding terms together:

$$\begin{aligned}
5|A|(4|A|C)^D \exp(-C\cdot\tilde{t}^2) + &|A|(|A|C)^D \exp(-C\lambda^2/2V^2_{\max}) && (73)\\
&\leq 5|A|(4|A|C)^D \exp(-C\cdot\tilde{t}^2) + |A|(|A|C)^D \exp(-C\cdot\tilde{t}^2) && (74)\\
&\leq 5|A|(4|A|C)^D \exp(-C\cdot\tilde{t}^2) + |A|(4|A|C)^D \exp(-C\cdot\tilde{t}^2) && (75)\\
&= 6|A|(4|A|C)^D \exp(-C\cdot\tilde{t}^2). && (76)
\end{aligned}$$

Therefore, we conclude that the Q-value concentration inequalities for both POMDP approximation error and particle belief approximation error are bounded by α_d, β_d at every node, respectively, with probability at least $1 - 6|A|(4|A|C)^D \exp(-C\cdot\tilde{t}^2)$.

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) The densities $\mathcal{Z}, \mathcal{T}, b_0$ have the property that, for any observation sequence $\{o_n\}^d_{n=1}$, the Rényi divergence of the target distribution $\mathcal{P}^d$ and sampling distribution $\mathcal{Q}^d$ (Eqs. (22) and (23)) is bounded above by $d^{\max}_\infty$ for all $d = 0, \ldots, D-1$:

$$d_\infty(\mathcal{P}^d || \mathcal{Q}^d) = \text{ess sup}_{x\sim\mathcal{Q}^d}\, w_{\mathcal{P}^d/\mathcal{Q}^d}(x) \leq d^{\max}_\infty \tag{77}$$

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

(iv) We can sample from the generating function G and evaluate the observation density $\mathcal{Z}$.

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

(vi) We restrict our analysis to all the beliefs $b \in B$ that are realizable from the initial belief b_0 through Bayesian updates with action sequences $\{a_n\}$ and observation sequences $\{o_n\}$.

Theorem 2 (Sparse Sampling-ω Coupled Optimality)**.** *Suppose conditions (i)-(vi) are satisfied. Then, for any $\lambda > 0$ and $0 < \delta \leq 1$, choosing particle count constant C that satisfies:*

$$C = \max \left\{ \left(\frac{4V_{\max} d_{\infty}^{\max}}{\lambda} \right)^2, \frac{64V_{\max}^2}{\lambda^2} \left(D \log \frac{24|A|^{\frac{D+1}{D}} V_{\max}^2 D}{\lambda^2} + \log \frac{1}{\delta} \right) \right\}, \tag{78}$$

the Q-function estimates $\hat{Q}^*_{\omega,d}(\bar{b}_d, a)$ obtained for all depths $d = 0, \ldots, D-1$, realized beliefs or histories b_d encountered in the Sparse Sampling-ω tree, and actions a are jointly near-optimal with respect to $Q^*_{\mathbf{P},d}$ and $Q^*_{\mathbf{M_P},d}$ with probability at least $1-\delta$:

$$|Q^*_{\mathbf{P},d}(b_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1-\gamma}, \tag{79}$$

$$|Q^*_{\mathbf{M_P},d}(\bar{b}_d, a) - \hat{Q}^*_{\omega,d}(\bar{b}_d, a)| \leq \frac{\lambda}{1-\gamma}. \tag{80}$$

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:

$$t_{\max}(\lambda, C) = \frac{\lambda}{4V_{\max} d_{\infty}^{\max}} - \frac{1}{\sqrt{C}} > 0 \tag{81}$$

$$\delta \geq 6|A|(4|A|C)^D \exp(-C \cdot \tilde{t}^2) \tag{82}$$

$$\tilde{t}_{\max}(\lambda, C) \equiv \max \left\{ t_{\max}(\lambda, C), \lambda / 4\sqrt{2} V_{\max} \right\} \tag{83}$$

Note that the constraint on $t_{\max}$ implies that the following must be true:

$$\frac{\lambda}{4V_{\max} d_{\infty}^{\max}} - \frac{1}{\sqrt{C}} > 0 \Longrightarrow C > \left(\frac{4V_{\max} d_{\infty}^{\max}}{\lambda} \right)^2, \tag{84}$$

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. (82). We note that due to the definition of $\tilde{t}_{\max}$, the following is true:

$$\tilde{t}_{\max}(\lambda, C) \geq \lambda / 4\sqrt{2} V_{\max}. \tag{85}$$

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

$$6|A|(4|A|C)^D \exp(-C \cdot \tilde{t}^2) \leq 6|A|(4|A|C)^D \exp(-C \cdot T) \tag{86}$$

$$\leq (24|A|^{\frac{D+1}{D}} C)^D \exp(-C \cdot T) \tag{87}$$

Consequently, if we show that Eq. (87) is bounded by δ, then we automatically show that the original quantity is bounded by δ as well. By defining $X \equiv 24|A|^{\frac{D+1}{D}}$, we want to show that this simplified formula is bounded above by δ:

$$\delta \geq (X \cdot C)^D \exp(-C \cdot T). \tag{88}$$

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

$$\frac{64V_{\max}^2}{\lambda^2}\left(D\log\frac{24|A|^{\frac{D+1}{D}}V_{\max}^2 D}{\lambda^2}+\log\frac{1}{\delta}\right) \Longrightarrow \frac{2}{T}\left(D\log\frac{XD}{T}+\log\frac{1}{\delta}\right) \tag{89}$$

Substituting in the second option of C:

$$(X \cdot C)^D \exp(-C \cdot T) = \delta^2 \frac{\left(\frac{2XD}{T}\log\frac{XD}{T}+\frac{2X}{T}\log\frac{1}{\delta}\right)^D}{\left(\frac{XD}{T}\right)^{2D}} \tag{90}$$

$$= \delta^2 \frac{\left(\frac{XD}{T}\log\left(\frac{XD}{T}\right)^2+\frac{XD}{T}\log\left(\frac{1}{\delta}\right)^{2/D}\right)^D}{\left(\frac{XD}{T}\right)^{2D}} \tag{91}$$

$$= \delta^2 \frac{\left(\log\left(\frac{XD}{T\delta^{1/D}}\right)^2\right)^D}{\left(\frac{XD}{T}\right)^D} = \delta\left(\frac{\log\left(\frac{XD}{T\delta^{1/D}}\right)^2}{\left(\frac{XD}{T\delta^{1/D}}\right)}\right)^D \tag{92}$$

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

$$(X \cdot C)^D \exp(-C \cdot T) \leq \delta. \tag{93}$$

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:

$$C = \max\left\{\left(\frac{4V_{\max}d_{\infty}^{\max}}{\lambda}\right)^2, \frac{64V_{\max}^2}{\lambda^2}\left(D\log\frac{24|A|^{\frac{D+1}{D}}V_{\max}^2 D}{\lambda^2}+\log\frac{1}{\delta}\right)\right\}. \tag{94}$$

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-ω policy is bounded by α_d for all d, a with probability at least $1-\delta$. Since $\alpha_d \leq \alpha_0$, the following holds for all $d = 0, \dots, D-1$ with probability at least $1-\delta$ through Lemmas 1 and 2:

$$|Q_{\mathbf{P},d}^*(b_d, a) - \hat{Q}_d^*(\bar{b}_d, a)| \leq \alpha_0 \leq \sum_{d=0}^{D-1}\gamma^d\lambda \leq \frac{\lambda}{1-\gamma}, \tag{95}$$

$$|Q_{\mathbf{M_P},d}^*(\bar{b}_d, a) - \hat{Q}_d^*(\bar{b}_d, a)| \leq \beta_0 \leq \sum_{d=1}^{D}\gamma^d\lambda \leq \frac{\lambda}{1-\gamma}. \tag{96}$$

□

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 ξ: $|Q^*(b,a) - \hat{Q}(b,a)| \leq \xi$. Let $V^\pi(b_0)$ denote the value of executing π starting at belief b_0 with an exact Bayesian belief update, $b_{t+1} = b_t ao$, between each call to the policy. Then*

$$V^*(b_0) - V^\pi(b_0) \leq \frac{2\xi}{1-\gamma}. \tag{97}$$

Proof. First, note that if an action is chosen by π, it must appear better than π^* 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

$$Q^*\left(b_t,\pi^*(b_t)\right) - \mathbb{E}[Q^*\left(b_t,\pi(b_t)\right)] \leq 2\xi. \tag{98}$$

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. (98):

$$\begin{aligned} V^*(b_{D-1}) - V^\pi(b_{D-1}) &= Q^*(b_{D-1},\pi^*(b_{D-1})) - \mathbb{E}[Q^\pi(b_{D-1},\pi(b_{D-1}))] && (99)\\ &= Q^*(b_{D-1},\pi^*(b_{D-1})) - \mathbb{E}[Q^*(b_{D-1},\pi(b_{D-1}))] && (100)\\ &\leq 2\xi. && (101) \end{aligned}$$

The inductive step is verified by subtracting and adding $\mathbb{E}[Q^*\left(b,\pi(b)\right)]$, using the bound in Eq. (98), and applying the inductive hypothesis:

$$\begin{aligned} V^*(b_t) - V^\pi(b_t) &= Q^*\left(b,\pi^*(b)\right) - \mathbb{E}[Q^\pi(b,\pi(b))] && (102)\\ &= \underbrace{Q^*\left(b,\pi^*(b)\right) - \mathbb{E}[Q^*\left(b,\pi(b)\right)]}_{\text{Bounded by Eq. (98)}} + \mathbb{E}[Q^*\left(b,\pi(b)\right)] - \mathbb{E}[Q^\pi(b,\pi(b))] && (103)\\ &\leq 2\xi + \mathbb{E}[Q^*\left(b,\pi(b)\right)] - \mathbb{E}[Q^\pi(b,\pi(b))] && (104)\\ &= 2\xi + \mathbb{E}[R\left(b,\pi(b)\right)] + \gamma\mathbb{E}[V^*(b_{t+1}))] - \mathbb{E}[R(b,\pi(b))] - \gamma\mathbb{E}[V^\pi(b_{t+1}))] && (105)\\ &= 2\xi + \gamma\mathbb{E}[V^*(b_{t+1})) - V^\pi(b_{t+1}))] && (106)\\ &= 2\xi + \gamma \sum_{d=t+1}^{D-1} \gamma^{d-t} 2\xi = \sum_{d=t}^{D-1} \gamma^{d-t} 2\xi. && (107) \end{aligned}$$

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

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

□

Theorem 4 (Particle Belief MDP Approximate Policy Convergence)**.** *Suppose a near-optimal MDP planning algorithm $\mathcal{A}$ is used to plan with particle belief MDP $\mathbf{M_P}$ repeatedly in a closed loop with POMDP environment $\mathbf{P}$ and an exact Bayesian belief updater to process observations from the environment. Further assume that regularity conditions (i)-(vi) are met for $\mathbf{M_P}$ and that $\mathcal{A}$ can approximate the Q-values of $\mathbf{M_P}$ with arbitrary precision $\varepsilon_{\mathcal{A}}$ with probability at least $1-\delta_{\mathcal{A}}$. Then, for any $\varepsilon > 0$, we can choose C such that the value obtained by planning with $\mathcal{A}$ in $\mathbf{M_P}$ is within ε of the optimal POMDP value function at b_0:*

$$V^*_{\mathbf{P}}(b_0) - V^{\mathcal{A}}_{\mathbf{M_P}}(b_0) \leq \varepsilon. \tag{108}$$

Proof. First, we choose λ and $\delta_{\mathbf{M_P}}$ for the particle belief MDP approximation to be the following, with $\varepsilon_{\mathbf{M_P}} = \frac{2\lambda}{1-\gamma}$ as per the definition in the proof of Theorem 3:

$$\lambda = \frac{(1-\gamma)^2}{8}\varepsilon - \frac{1-\gamma}{2}\varepsilon_{\mathcal{A}} \Longrightarrow \varepsilon_{\mathbf{M_P}} = \frac{1-\gamma}{4}\varepsilon - \varepsilon_{\mathcal{A}}, \tag{109}$$

$$\delta_{\mathbf{M_P}} = \frac{\varepsilon_{\mathbf{M_P}} + \varepsilon_{\mathcal{A}}}{V_{\max} D(1-\gamma)} - \delta_{\mathcal{A}}. \tag{110}$$

In this context, we mathematically mean a near-optimal MDP planning algorithm $\mathcal{A}$ to be one that can obtain arbitrarily small values of $\varepsilon_{\mathcal{A}}, \delta_{\mathcal{A}}$ that would satisfy $\lambda > 0$ and $0 < \delta_{\mathbf{M_P}} \leq 1$. Then, we can choose C through Theorem 3 such that we can invoke Corollary 1 to obtain Q-value estimation accuracy $\xi = \varepsilon_{\mathbf{M_P}} + \varepsilon_{\mathcal{A}}$ with worst case probability $\delta' = \delta_{\mathbf{M_P}} + \delta_{\mathcal{A}}$. Consequently, with our choice of λ and $\delta_{\mathbf{M_P}}$ above, ξ and δ' are equal to the following:

$$\xi = \frac{1-\gamma}{4}\varepsilon, \tag{111}$$

$$\delta' = \frac{\xi}{V_{\max} D(1-\gamma)}. \tag{112}$$

During policy execution, we create a new independent tree and choose an action based on the estimated Q-values. This means with algorithm $\mathcal{A}$ equipped with particle belief states, there is at most δ' probability that $|Q^*_{\mathbf{P},0}(b_0,a) - \hat{Q}^{\mathcal{A}}_{\mathbf{M_P},0}(\bar{b}_0,a)| > \xi$ at each of the D steps of the POMDP.

Thus, with probability at least $1 - D\delta'$, we execute a policy that meets the assumptions of Lemma 3A with ξ, and hence by Lemma 3A, the difference between the optimal value and the average accumulated reward for this case is at most $\frac{2\xi}{1-\gamma}$. 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 $2V_{\max}$ from the optimal policy. Combining these two cases, we have

$$\begin{aligned} V^*_{\mathbf{P}}(b_0) - V^{\mathcal{A}}_{\mathbf{M_P}}(b_0) &\leq (1-D\delta')\frac{2\xi}{1-\gamma} + D\delta'(2V_{\max}) && (113)\\ &\leq \frac{2\xi}{1-\gamma} + D\delta'(2V_{\max}) && (114)\\ &= \frac{2\xi}{1-\gamma} + \frac{2\xi}{1-\gamma} && (115)\\ &= \varepsilon. && (116) \end{aligned}$$

□

Appendix F. Experiment Details

	c_{UCB}	β_{UCB}	k_a	α_a	k_o	α_o	$m_{\min}$	δ	C	Depth
Laser Tag (D, D, D)										
Sparse-PFT	15	0.22	-	-	15	-	-	-	96	37
PFT-DPW	25	0.09	-	-	5	0.33	-	-	25	48
POMCPOW	26	-	-	-	4	0.03	-	-	-	50
AdaOPS	-	-	-	-	-	-	30	0.1	-	90
POMCP	26	-	-	-	-	-	-	-	-	50
QMDP	-	-	-	-	-	-	-	-	-	-
Light Dark (D, D, C)										
Sparse-PFT	95	0.39	-	-	24	-	-	-	134	28
PFT-DPW	93	0.30	-	-	13	0.08	-	-	33	20
POMCPOW	90	-	-	-	5	0.07	-	-	-	20
AdaOPS	-	-	-	-	-	-	30	0.1	-	90
POMCP	83	-	-	-	-	-	-	-	-	20
QMDP	-	-	-	-	-	-	-	-	-	-
Sub Hunt (D, D, C)										
Sparse-PFT	20	0.25	-	-	27	-	-	-	23	20
PFT-DPW	85	0.08	-	-	10	0.08	-	-	79	20
POMCPOW	17	-	-	-	6	0.01	-	-	-	50
AdaOPS	-	-	-	-	-	-	30	0.1	-	90
POMCP	17	-	-	-	-	-	-	-	-	84
QMDP	-	-	-	-	-	-	-	-	-	-
VDP Tag (C, C, C)										
Sparse-PFT	16	0.12	28	-	28	-	-	-	385	33
PFT-DPW	23	0.25	22	0.32	21	0.04	-	-	132	44
POMCPOW	110	-	30	0.03	5	0.01	-	-	-	10
VDP Tag^D (C, D, C)										
Sparse-PFT	76	0.08	-	-	25	-	-	-	444	46
PFT-DPW	10	0.18	-	-	9	0.11	-	-	330	22
POMCPOW	31	-	-	-	5	0.05	-	-	-	10
AdaOPS	-	-	-	-	-	-	40	0.25	-	90

Table 2: Summary of hyperparameters used in experiments.

For UCT methods, we vary c_{UCB}, the UCB exploration parameter, and β_{UCB}, the polynomial UCB factor. k_a and α_a are action progressive widening parameters, where new actions are added if widening criterion $|C(h)| \leq k_a N(h)^{\alpha_a}$ is met. Similarly, k_o and α_o are observation progressive widening parameters, where new actions are added if widening criterion $|C(ha)| \leq k_o N(ha)^{\alpha_o}$ is met. Sparse-PFT uses $\alpha_a = \alpha_o = 0$. C is the number of particles constituting internal tree beliefs for PFT methods. m_{min} is the minimum number of particles required to approximate a belief for AdaOPS. Finally, δ is the maximum distance distance between beliefs resulting from observation branches required to merge the branches for AdaOPS.

	$\hat{V}$	L_0	U_0
Laser Tag (D, D, D)			
Sparse-PFT	QMDP PO-Rollout	-	-
PFT-DPW	QMDP PO-Rollout	-	-
POMCPOW	FO-Value	-	-
AdaOPS	-	Random Rollout	QMDP
POMCP	Random Rollout	-	-
QMDP	-	-	-
Light Dark (D, D, C)			
Sparse-PFT	QMDP PO-Rollout	-	-
PFT-DPW	QMDP PO-Rollout	-	-
POMCPOW	FO-Value	-	-
AdaOPS	-	Random Rollout	QMDP
POMCP	Random Rollout	-	-
QMDP	-	-	-
Sub Hunt (D, D, C)			
Sparse-PFT	QMDP PO-Rollout	-	-
PFT-DPW	QMDP PO-Rollout	-	-
POMCPOW	FO-Value	-	-
AdaOPS	-	Random Rollout	QMDP
POMCP	Random Rollout	-	-
QMDP	-	-	-
VDP Tag (C, C, C)			
Sparse-PFT	Random Rollout	-	-
PFT-DPW	Random Rollout	-	-
POMCPOW	Random Rollout	-	-
VDP Tag^D (C, D, C)			
Sparse-PFT	Random Rollout	-	-
PFT-DPW	Random Rollout	-	-
POMCPOW	Random Rollout	-	-
AdaOPS	-	Random Rollout	10^6
POMCP	Random Rollout	-	-

Table 3: Summary of leaf node value estimators used in experiments.

QMDP PO-Rollout corresponds to sampling a "true state" from a leaf node particle belief and simulating the state/belief dynamics with a particle filter as a belief updater and QMDP as a policy.

The returns following the trajectory of the sampled “true state” are taken as a value estimate for the leaf node. FO-Value (”Fully Observable Value”) corresponds to using the MDP value for the state representation of a particle. QMDP corresponds to using the belief value estimate given by a QMDP policy. Random Rollout corresponds to sampling a state from a particle belief and simulating it forward using a random policy. The returns of this simulation are used as the initial leaf node value estimate. A constant number (e.g. 10^6) indicates a belief-independent static initial value estimate. UCT solvers only require a single value estimate ($\hat{V}$), whereas AdaOPS requires lower and upper bounds on belief value, L_0 and U_0 respectively.

References

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

Bai, H., Hsu, D., & Lee, W. S. (2014). Integrated perception and planning in the continuous space: A POMDP approach. *International Journal of Robotics Research*, *33*(9), 1288–1302.

Bjarnason, R., Fern, A., & Tadepalli, P. (2009). Lower bounding Klondike solitaire with Monte-Carlo planning, In *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, In *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.

Couëtoux, A., Hoock, J.-B., Sokolovska, N., Teytaud, O., & Bonnard, N. (2011). Continuous upper confidence trees, In *Learning and intelligent optimization*
double progressive widening.

Crisan, D., & Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, *50*(3), 736–746.

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, In *ArXiv pre-print*.

Du, S. S., Hu, W., Li, Z., Shen, R., Song, Z., & Wu, J. (2021). When is particle filtering efficient for planning in partially observed linear dynamical systems?, In *Uncertainty in artificial intelligence*.

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, In *Robotics: Science and systems*.

Hoerger, M., & Kurniawati, H. (2021). An on-line POMDP solver for continuous observation spaces, In *Ieee international conference on robotics and automation (icra)*.

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.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning, In *European conference on machine learning (ecml)*.

Kurniawati, H., & Yadav, V. (2016). An online POMDP solver for uncertainty planning in dynamic environment. In *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, In *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, In *Ieee conference on decision and control (cdc)*.

Luo, Y., Bai, H., Hsu, D., & Lee, W. S. (2019). Importance sampling for online planning under uncertainty. *International Journal of Robotics Research*, *38*(2-3), 162–181.

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, In *Aaai conference on artificial intelligence (aaai)*.

Metelli, A. M., Papini, M., Faccio, F., & Restelli, M. (2018). Policy optimization via importance sampling, In *Advances in neural information processing systems (nips)*.

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, In *Ieee international conference on robotics and automation (icra)*.

Shah, D., Xie, Q., & Xu, Z. (2022). Nonasymptotic analysis of Monte Carlo tree search. *Operations Research*, *0*(0), https://doi.org/10.1287/opre.2021.2239.

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, In *Advances in neural information processing systems*.

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, In *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.