OO-POMCP: Robust Multi-Object Planning for Object-Oriented POMDPs

Arthur Wandzel[1] , **Seungchan Kim**[1] , **Stefanie Tellex**[1] and **Yoonseon Oh**[2]
[1]Department of Computer Science, Brown University
[2]Korea Institute of Science and Technology, South Korea
{arthur_wandzel, seungchan_kim, stefanie_tellex}@brown.edu, yoonseon_oh@kist.re.kr

Abstract

High-level human cognition supports reasoning about multiple objects as evident in manipulation or navigation tasks. Object-based reasoning in real-world environments, however, impose a challenge: as the number of considered objects scale, planning becomes increasingly computationally intractable. In this paper, we theoretically and empirically prove the efficiency of Object-Oriented Partially Observable Monte-Carlo Planning (OO-POMCP), an object-oriented online POMDP planner that robustly scales with the number of considered objects. The performance of OO-POMCP generalizes to a class of POMDPs explored in this paper where the model is factored in terms of objects. We first prove the sample efficiency of belief factorization via Rademacher complexity by deriving general upper bounds on the number of samples for online POMDP planning. We second evaluate OO-POMCP on domains relevant for multi-object reasoning and show that the performance of OO-POMCP is robust to the number of objects, even when considering low sample sizes, local spacial dependencies, and large observation spaces.

Introduction

High-level human cognition supports reasoning about multiple objects as evident in manipulation or navigation tasks. In parallel with advances in object-segmentation and object-recognition techniques for images, recently, object-based reasoning has received attention in robotics as a means of supporting rich interactions with the environment [Janner *et al.*, 2018; Devin *et al.*, 2018; Greff *et al.*, 2019; Li *et al.*, 2016; Pajarinen and Kyrki, 2017]. In real-world settings, however, the number of considered objects may scale arbitrarily, which makes planning increasingly computationally intractable. The unstructured and uncontrolled nature of complex real-world environments makes it essential to guarantee the performance of multi-object planning that is robust to the number of objects: whether it be for avoiding numerous pedestrians in a busy walkway or searching for the many survivors in a disaster site.

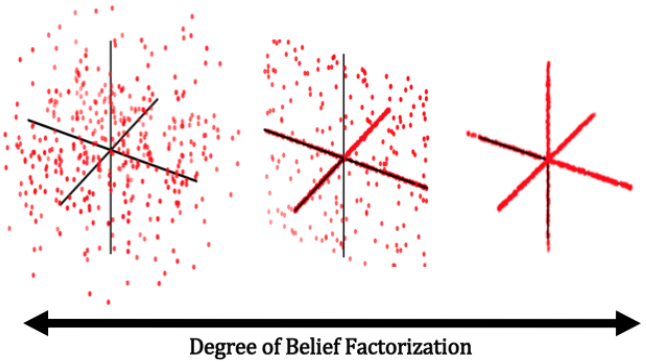

Figure 1: Visualization of 100 samples drawn from different belief representations varying from unfactored to factored, each dimension reflecting possible states of an object (3 in total). In this work, we analyze OO-POMCP—an object-oriented online POMDP planner that leverages the belief representation for robust multi-object planning.

Partially observable Markov decision processes (POMDPs) [Kaelbling *et al.*, 1998] have been popularly explored in sequential decision making in robotics by accounting for uncertainty as a consequence of a lack of knowledge of the full state of the environment (partial observably) or imprecise sensors or actuators (stochastic dynamics). However, POMDPs are intractable to exactly solve. The two sources of intractability have been termed the *curse of dimensionality* and the *curse of history* [Pineau *et al.*, 2006], which are defined respectively as: a POMDP planner reasons in terms of probability distributions over the state, called *beliefs*, and a POMDP planner must plan over action-observation contingencies, called *histories*, which increase double exponentially in depth of the planning horizon.

In multi-object planning, the size of the belief grows exponentially with the number of considered objects by representing object-to-object dependencies (see Figure 1). When reasoning about complex interactions, however, one cognitive strategy employed by humans is to separate each source of variation into "conceptual chunks" [Halford *et al.*, 1998].

In this paper, we theoretically and empirically prove the efficiency of OO-POMCP: an object-oriented online POMDP planing algorithm introduced in [Wandzel *et al.*, 2019]. We examine OO-POMCP in the context of object-oriented POMDPs (OO-POMDPs) which represent the state, transition, and observation spaces in terms of *classes* and *objects* [Wandzel *et al.*, 2019]. Similar to conceptual chunking, OO-POMCP factors the belief into independent object-specific distributions, which in turn reduces the computational complexity of POMDP planning.

We first theoretically analyze OO-POMCP by deriving bounds on the number of samples to estimate the Q-value function within online POMDP planning. The bounds make use of a Rademacher complexity measurement [Bartlett and Mendelson, 2002] taken from computational learning theory, which is sensitive to reward signal variance [Jiang *et al.*, 2015]. We establish a direct relationship between the belief representation and the number of samples required for POMDP planning by comparing bounds while varying the degree of belief factorization in terms of objects. We demonstrate that factorization serves as a complexity control parameter of the POMDP model. This result validates the sample efficiency of OO-POMCP on a class of POMDPs where the model can be represented in terms of object factors.

We second empirically evaluate OO-POMCP on a number of domains relevant for multi-object planning, comparing performance to POMCP a well known online planning algorithm for large domains [Silver and Veness, 2010]. Across a class of POMDPs, OO-POMCP provides robust performance as the number of objects scale while preserving sample efficiency versus the existing POMCP algorithm. OO-POMCP, furthermore, deals well with large jointly considered observation spaces and local spatial dependencies between objects. Taken together, these theoretical and empirical results prove the efficiency of OO-POMCP as well as lay the foundation for an exciting line of research for future online planning algorithms that explores dynamically factoring the belief on task onset.

Related Work

Online POMDP planning has demonstrated high performance over other approaches (e.g. offline planning) while scaling to large domains [Ross *et al.*, 2008]. State-of-the-art POMDP solvers, DESPOT [Somani *et al.*, 2013] or POMCP [Silver and Veness, 2010], are online planners that break both intractability curses via Monte Carlo sampling. These algorithms interleave planning and action execution within a *planning cycle* composed of two steps: constructing a forward search tree from the current belief for planning an action (*search*) and updating the current belief to the next belief so as to incorporate the observation yielded from executing the planned action (*tracking*).

Although shown to compute near-optimal policies for large POMDPs, these algorithms are susceptible to tracking error by estimating the next belief. This error compounds over the number of planning cycles and with size of the state, action, or observation spaces [Sunberg and Kochenderfer, 2018; Garg *et al.*, 2019]. Belief estimation, furthermore, is exceptionally challenging in a multi-object setting because the size of the belief increases exponentially with the number of considered objects. OO-POMCP differs from existing methods by representing the belief in terms of object factors rather than jointly. Furthermore, OO-POMCP separates search from tracking into two dedicated processes. Together this allows OO-POMCP to tractably update the belief per planning cycle so as to support error-free tracking.

A factored belief representation has been explored as a means of supporting tractable POMDP planning with analysis [Poupart, 2005; Guestrin *et al.*, 2001; McAllester and Singh, 2013]. However, not in the context of Monte Carlo online planning approaches, which is the current state-of-the-art for probabilistic POMDP planning. Our analysis of OO-POMCP, furthermore, defines bounds in terms of samples with an additional Rademacher complexity measurement, which constitutes the first ever application to POMDPs. In addition to a factored representation, other orthogonal approaches for approximate POMDP planning has looked at compressed beliefs [Roy *et al.*, 2005] or beliefs defined over only components of the state relevant for decision making [Ong *et al.*, 2010].

One algorithm similar to OO-POMCP is Factored-Value POMCP (FV-POMCP) by [Amato and Oliehoek, 2015]. FV-POMCP factors the problem in terms of agents within a multiagent POMDP. One algorithmic difference is FV-POMCP maintains separate forward search trees per factor which introduces additional hyperparameters as each factor requires a weight to return a joint action. OO-POMCP avoids this by jointly considers actions and observations over objects, which eliminates hyperparameter tuning, however, at the cost of inducing a larger forward search tree.

Preliminaries

A POMDP is defined as a 7-tuple composed of the state space S, action space A, observation space Ω, transition function $T(s', a, s) = p(s'|s, a)$, reward function $R(s, a) : S \times A \rightarrow \mathbb{R}$ with probability $p(r|s, a)$, γ discount factor, and observation function $O(o, s', a) = p(o|s', a)$:

$$\langle S, A, T, R, \gamma, \Omega, O \rangle.$$

The agent maintains a belief, $b \in B$, which is a probability distribution over the state space, such that the agent believes it is in state s with probability $p(s)$. The POMDP reward function defined on beliefs returns the expected reward under b: $R(b, a) = \sum_s b(s)R(s, a)$, Executing an action a yields an observation o, which is used to compute a new belief via an update function (with η normalizing constant):

$$b(s') = \eta\, O(o, s', a) \sum_{s \in S} T(s', a, s)\, b(s). \tag{1}$$

Rademacher complexity

Rademacher complexity measures the difficulty of estimating a function of a target function class from samples drawn from the training set distribution [Bartlett and Mendelson, 2002]. We can compute the *Rademacher empirical average* of a class

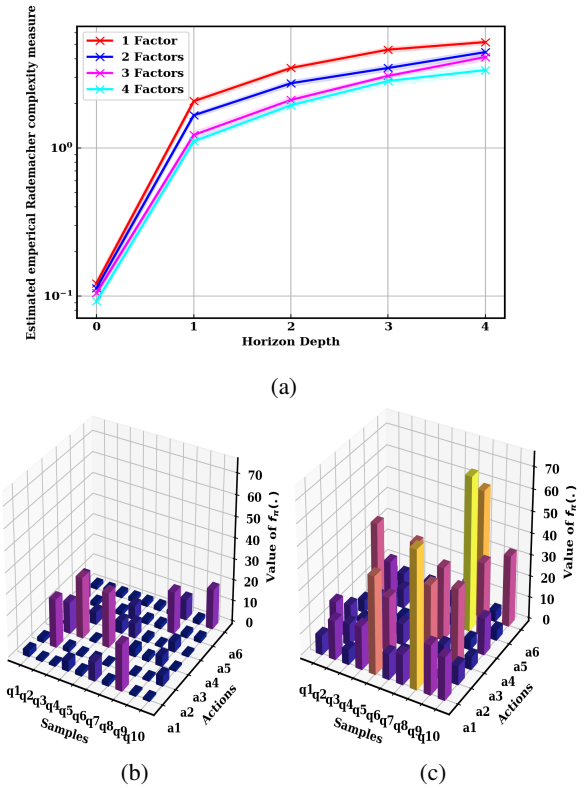

Figure 2: (a). The relationship between the estimated empirical Rademacher complexity over horizon depth with number of factors of the POMDP model: 4 is fully factored and 1 is unfactored for all objects. Visualization of $f_\pi(.)$ evaluation at horizon depth 4 for a fully factored (b). and unfactored (c). model for 10 samples per action; each sample averaged over 20 policies. Note: the difference in degree of value variation.

of functions $\mathcal{F}$ with respect to a training set $X = \{x_1, ..., x_k\}$ drawn from distribution $\mathcal{D}$ as follows:

$$\widehat{\mathfrak{R}}_k(\mathcal{F}, X) = \mathop{\mathbb{E}}_{\sigma}\left[\sup_{f\in\mathcal{F}} \frac{1}{k}\sum_{i=1}^{k}\sigma_i f(x_i)\right], \tag{2}$$

where the expectation is taken over the distribution of the Rademacher variables $\sigma = \{\sigma_1, ..., \sigma_k\}$. Each Rademacher variable σ_i is an independent random variable with value 1 or -1 with probability 1/2. The Rademacher empirical average measures the richness of function class $\mathcal{F}$ by recording how well on average the best fitting function in the function class correlates to random noise of the Rademacher variables over training set X. If there is no variance over $f \in F$, that is, each function produces the same constant value, then $\widehat{\mathfrak{R}}_k(.) = 0$.

Online POMDP Planning

The aim of planning is to return an optimal policy $\pi^* \in \Pi$ evaluated with respect to a model M of the environment [Sutton and Barto, 2018]. In online POMDP planning, the planner employs local forward search from the current belief to beliefs reachable within a planning horizon of H, denoted as: $\bar{B}$. We term a policy defined over $\bar{B}$, $\pi : \bar{B} \rightarrow A$, a policy tree reflecting an assignment of actions for reachable beliefs. The online POMDP planning problem consists of selecting an action maximizing the Q-value function from the current belief b: measures the expected reward of executing an action a in belief b while following policy tree π.

The Q-value function for a POMDP can be written as a piece-wise linear function of the state: $Q_\pi(b,a) = \sum_s b(s)V_\pi^a(s)$ [Kaelbling *et al.*, 1998], reflecting that the belief can be represented as a vector of probabilities of length $|S|$. We define $V_\pi^a(s)$ and $V_\pi^{ao}(s)$ respectively as the expected value of state s after taking action a or taking a and receiving o, and thereafter following policy tree π. The following is a novel explicit formulation of the POMDP Q-value function and POMDP model:

$$Q_\pi(b,a) = \sum_s b(s)V_\pi^a(s) \tag{3}$$

$$= \sum_s b(s)[R(s,a) + \gamma\sum_{s'}T(s'|s,a)\sum_o O(o|s',a)V_\pi^{ao}(s')]$$

$$= \sum_s p(s)[\sum_r rp(r|s,a) + \gamma\sum_{s',o}p(s'|s,a)p(o|s',a)V_\pi^{ao}(s')]$$

$$= \sum_s p(s)[\sum_r rp(r|s,a) + \gamma\sum_{s',o}p(o,s'|s,a)V_\pi^{ao}(s')]$$

$$= \sum_s \sum_{r,o,s'} p(s)[rp(r,o,s'|s,a) + \gamma p(r,o,s'|s,a)V_\pi^{ao}(s')]$$

$$= \sum_s \sum_{r,o,s'} p(s)p(r,o,s'|s,a)[r + \gamma V_\pi^{ao}(s')]. \tag{4}$$

Line 2 is from [Kaelbling *et al.*, 1998] and proof of line 4 is given in Appendix C. The form of Equation 4 reveals the complexity of POMDP online planning: the first probability term $p(s)$ corresponds to the belief and the *curse of dimensionality* while the second $p(r,o,s'|s,a)$ to the POMDP model and the *curse of history*.

Factored OO-POMDP

The OO-POMDP is a framework for representing uncertainty over objects [Wandzel *et al.*, 2019]. More exactly, an object is a state abstraction with a semantic reference. As a generalization of OO-MDPs [Diuk *et al.*, 2008], there are two levels of organization: classes and objects. An OO-POMDP is formally defined as a 10-tuple:

$$\langle \mathcal{C}, \textsc{Att}(c), \textsc{Dom}(a), Obj, A, T, R, \gamma, \Omega, O\rangle.$$

States are represented as an $|Obj|$ length vector of object states such that $s = \langle s_1, ..., s_n\rangle$ and $S = \{S_1 \times ... \times S_n\}$, for a state and the state space respectively, where S_i is the state subspace of the ith object. This representation is akin to a factored-state MDP [Strehl *et al.*, 2007], however, there is additional object-specific structure. Each object is an instance of a particular class, $c \in \mathcal{C}$ consisting of a set of class-specific attributes $Att(c)$. Each attribute has a domain of possible values $Dom(a)$. All possible values of an object's attributes make up the object subspace S_i (e.g. all (x,y) locations that object i can occupy). This additional structure of OO-POMDPs allows for uncertainty to be represented with varying degrees of specificity: over a class of objects, a single object, or an object-attribute.

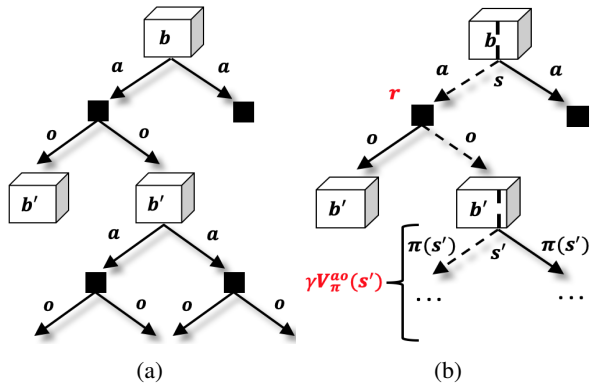

Figure 3: (a). Forward search tree from the current belief Note: not all belief nodes were drawn for space constraints. (b). $Q_\pi(b,a)$ backup diagram for a sample drawn from $p(s)p(r,o,s'|s,a)$ (dashed lines) for a given action and π. Accumulated reward denoted in red.

Factored OO-POMDP Model

In an OO-POMDP, the belief can be factored and compactly written as:

$$b(s) = \prod_{i=1}^{|C|} b(c_i), \tag{5}$$

where $C = \{c_1, ..., c_m\}$ is a factorization of m non-overlapping *object-factors* c_i and each c_i is a grouping of object substates (e.g. $c_i = \langle s_1, s_2 \rangle$, $c_i \in S_1 \times S_2$). One implication of a factored belief is that the OO-POMDP model is factored such that the reward, observation, and transition functions are defined in terms of independent object-factors.

The Q-value function for a factored model, furthermore, may be decomposed as a linear function defined in terms of each object-factor, $Q_\pi(b,a) = \sum_{i=1}^{|C|} Q_\pi(b_i, a)$:

$$\begin{aligned}
&Q_\pi(b,a) = \sum\nolimits_s b(s) V_\pi^a(s) \\
&= \sum\nolimits_{c_1} \sum\nolimits_{\bar{s}} b(\bar{s}|c_1) b(c_1) [V_\pi^a(\bar{s}) + V_\pi^a(c_1)] \\
&= \sum\nolimits_{\bar{s}} b(\bar{s}) \sum\nolimits_{c_1} b(c_1) [V_\pi^a(\bar{s}) + V_\pi^a(c_1)] \\
&= \sum\nolimits_{\bar{s}} b(\bar{s}) V_\pi^a(\bar{s}) + \sum\nolimits_{c_1} b(c_1) V_\pi^a(c_1) \\
&= \sum\nolimits_{\bar{s}} b(\bar{s}) V_\pi^a(\bar{s}) + Q(b_1, a)
\end{aligned}$$

Line 2: $\bar{s} = s \setminus c_1$; Line 3: $\bar{s}$ is independent of c_1; Line 4: c_1 is only in the domain of $V_\pi^a(c_1)$; Our final equation is achieved by repeating the above steps for $|C|$. This proof is satisfied by assuming a linear value function over objects such that for all s: $V(s) = V(c_1) + ... + V(c_{|C|})$. Note: the original Q-value function is retained when $|C| = 1$. With i indexing an object, the Q-value function for a factored belief b_i is expressed as:

$$\begin{aligned}
&Q_\pi(b_i, a) \\
&= \sum_{c_i} \sum_{r_i, o_i, c_i'} p(c_i) p(r_i, o_i, c_i' | c_i, a) (r_i + \gamma V_\pi^{ao_i}(c_i')).
\end{aligned} \tag{6}$$

How the OO-POMDP is factored may vary from 1 to n factors, n being the total number of considered objects. We define M_C to be the POMDP model defined according to C. $M_{|C|=1}$ implies a POMDP model that represents all object-to-object dependencies and $M_{|C|=n}$ implies a model that does not represent any object-to-object dependencies. While $M_{|C|=1}$ models are the most general, as we prove below, these models are disadvantageous to plan in for large n.

OO-POMCP

OO-POMCP [Wandzel *et al.*, 2019] extends POMCP [Silver and Veness, 2010]. These algorithms apply Monte Carlo Tree Search (MCTS) to POMDPs by constructing a forward search tree per planning cycle for action selection. In the tree, each node is a belief (the root is the current belief) and each branch is a possible action-observation contingency (see Figure 3a). A Monte Carlo *simulation* consists of sampling a state from the current belief that follows a sampled action-observation path down the forward search tree. The Q-value function is estimated as the average of discounted returns and the belief as a multiset (implicitly capturing the frequency) of states encountered at each node. Previous algorithms, like POMCP, perform a filtering step to track the belief: only the node corresponding to the next real-world action and observation is selected as the next belief. If there are too few particles in this node, the performance of the planner declines, as a consequence of belief underestimation, which compounds over planning cycles. OO-POMCP, instead, represents the belief in terms of object factors and separates search from tracking into two dedicated processes, thereby supporting explicit belief updates for error-free tracking.

Theoretical Analysis

In this section, we derive sample bounds for estimating the Q-value function of an OO-POMDP. We show how these bounds vary when factoring the model in terms of objects. The first part defines the online planning problem and the POMDP model. The second part defines the factored OO-POMDP model and class of POMDPs targeted by OO-POMCP. The third and final part derives the bounds over the class of POMDPs via Rademacher complexity, demonstrating the the class of POMDPs targeted by OO-POMCP is the most sample efficient for POMDP online planning.

Proof of Q-Value Estimation Error Bound

We investigate how factorization improves the sample efficiency of POMDP planning by comparing bounds on the number of samples to estimate the Q-value function. Specifically, we bound the quantity $Q_\pi(b,a) - \widehat{Q}_\pi(b,a)$ for a fixed b while varying the number of factors of the OO-POMDP model $|C|$. Our bound utilizes a Rademacher complexity measure, as inspired by [Jiang *et al.*, 2015], which offers tight bounds due to its dependence on the underlying sample distribution [Mitzenmacher and Upfal, 2017].

Theorem 1. *Let M_C be the OO-POMDP model factored in a set of C object-factors and the reward function is non-negative with maximum value R_{max}. Let $b_i = p(c_i)$ be the*

factored belief defined over one such object-factor. Then for all b_i reflecting all $c_i \in C$ and $K = \sum_{i=1}^{|C|} \sum_{c_i} c_i$:

$$\max_{\pi:\bar{B}\to A} \left| Q_\pi(b,a) - \widehat{Q}_\pi(b,a) \right| \leq$$

$$2 \max_{\substack{c_i \in b_i \forall i \\ a \in A}} \widehat{\mathfrak{R}}_{T_{c_i,a}}(F) + \frac{3R_{max}}{1-\gamma}\sqrt{\frac{1}{2n}\log\frac{4|A|K}{\delta}}, \quad (7)$$

with probability $\geq 1-\delta$ for all $f \in \mathcal{F}$, where

- $\mathcal{F} = \{f_\pi : \pi \in \bar{B} \to A\}$*, with*
  $$f_\pi(r_i, o_i, c_i') = r_i + \gamma V_\pi^{ao_i}(c_i').$$
- $T_{c_i,a}$ *is a training set of n reward, observation, and next-state pairs drawn from the true distribution $p(r_i, o_i, c_i'|c_i, a)$ with respect to a c_i drawn from $p(c_i)$ and fixed action a.*

The proof of Theorem 1 as well as the experiment details of each chart in this section is given in Appendix A and B. The result is a bound on the error of the Q-value estimate over all policy trees with respect to the current belief. We compare the bound across OO-POMDP models, M_C, varying the degree of factorization from complete independence $|C| = n$ to complete interdependence $|C| = 1$.

The first term reflects the complexity of the OO-POMDP models in evaluating the Q-value function. Rademacher complexity is sensitive to the variance of $f_\pi(.)$ evaluations over all π of the policy space. Each function is evaluated for an action and sample (c_i, r_i, o_i, c_i') for a randomly selected object-factor i, following a policy tree π (see Figure 3b).

Figure 2a compares the estimated empirical Rademacher complexity measurement over horizon depth. The measurement increases with the horizon depth reflecting more variance of $f\pi(.)$. At horizon 0, for example, each evaluation can differ at most by r_i. The measurement increases, more importantly, with the degree of factorization of the OO-POMDP model, which holds constant when the horizon is greater than 0. Figure 2b and 2c visualizes the variance of evaluations of $f_\pi(.)$ for a completely factored versus unfactored OO-POMDP model at horizon 4. More variance over the policy space correlates with a higher Rademacher score reflecting the capacity to fit random noise of the Rademacher variables. Qualitatively, we can see greater variance for the policy spaces represented under the unfactored versus factored model. These charts validate that factorization serves as a complexity control parameter of the POMDP model as measured by Rademacher complexity.

The second term is sensitive to the size of the belief K which counts all the states in the object-factors. Each state represents an estimated dimension in the belief. Intuitively, the size of K increases exponentially with the number of objects jointly represented in the belief. Varying the factorization results in K being $\sum_i |S_i|$ versus $\prod_i |S_i|$ assuming equally sized subspaces per object-factor. The exponential growth of the dimensionality of the belief is a direct consequence of representing object-to-object dependencies in the joint state. Factorization, thus, mediates a trade off between generality and tractability: by not representing dependencies, a factored belief requires less dimensions to estimate. Collectively, the upper bound validates that factoring the OO-POMDP model in terms of objects, as performed by OO-POMCP, requires fewer samples for estimating the Q-value function for online POMDP planning.

Empirical Evaluation

In this section, we empirically evaluate OO-POMCP on a number of domains relevant for multi-object planning: Rock Sample (*RS*), Joint Rock sample (*Joint RS*), Multi-Object Search (*MOS*), and Car Driving Among Pedestrians (*CDAP*). The POMDPs represented by these domains are members of a class of POMDPs where the model can be factored in terms of objects without model inaccuracy: the reward, observation, and transition functions can be represented in terms of independent objects. We compare against various benchmark conditions including POMCP. We do not include DESPOT as a comparison method because both methods are complementary improvements upon POMCP: while the focus of OO-POMCP is on tracking, DESPOT focuses on search. The following table compares the size of the belief and state-, observation-, and action-spaces of each domain (* denotes approximate):

Domain	Belief	State	Observation	Action
RS(11,11)	10^3*	10^5*	2	15
Joint RS(11,11)	10^3*	10^5*	10^3*	5
MOS	$10^{21}*$	$10^{27}*$	10^9*	$12*$
CDAP	$10^{14}*$	$10^{12}*$	$10^{14}*$	3

Rock Sample

Rock Sample(m,n) is a canonical POMDP domain for benchmarking algorithms [Smith and Simmons, 2004]. The objective is that an agent must collect n rocks in a m by m map. The belief is defined over each rock, whether it is good or bad. The agent received positive reward for sampling a good rock $+10$, as well as exiting the map on the far East side $+10$, and additionally -10 for sampling a bad rock. Each target rock can be sensed, yielding an observation, with a accuracy decreasing exponentially in the distance between the agent and the object. The actions are $n+5$, n `CHECK` actions for sensing, 4 `MOVE` actions in each cardinal direction, and 1 `SAMPLE` action.

In our experiment, we compare the discounted cumulative reward ($\gamma = .95$) of OO-POMCP against POMCP on Rock Sample with $m = 11$ and $n = 11$ while varying the number of Monte Carlo *simulations*. Each point is the average of 100 trials. This experiment evaluates the sample efficiency of OO-POMCP. We add another condition to test performance in a large observation space by considering a single `CHECK` action that *jointly* observes all rocks, which should improve performance by increasing observational information per `CHECK` action. We see that OO-POMCP outperforms POMCP even for a low number of simulations and the margin of difference increases when jointly considering observations. This is because OO-POMCP to successfully track the belief, invariant to the size of the observation space. More specifically, when POMCP receives a rare observation (one that is not estimated)

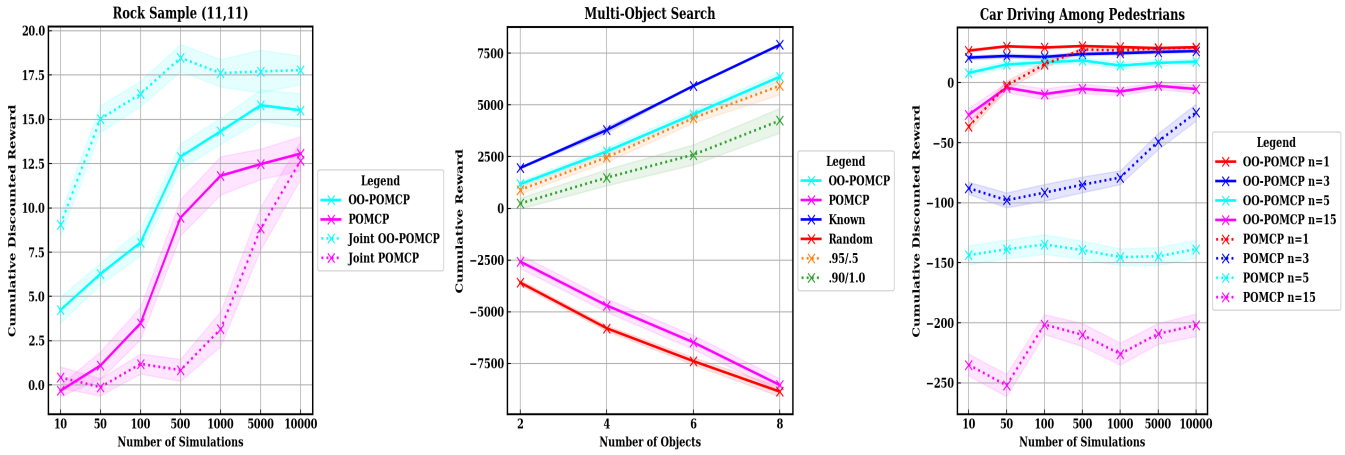

Figure 4: Algorithm performance across three domains comparing OO-POMCP with POMCP.

the next belief node will not have states to represent the next belief. When this occurs, performance fails.

Multi-Object Search

In the Multi-Object Search domain [Wandzel *et al.*, 2019], the goal is for the agent to locate the 2-D coordinate of n objects in a roomed, indoor environment with l possible locations. The belief is defined over the configuration of objects in locations such that its dimensionality is l^n ($l = 500$), constituting a very challenging POMDP. The reward function is $+1000$ and -1000 for correctly and incorrectly locating an object and -10 for all other actions with an additional cost for movement actions proportional to distance traveled. The available actions are: `LOOK` in each cardinal direction, `FIND` marks a location of an object, `MOVE` moves the agent determinsitically to the center of each room or within room locations via a variable connection topological map. `LOOK` yields an observation indicating the presence of objects in any location within a fan-shaped sensor region.

We test if OO-POMCP can exploit local spatial-dependencies between objects by distributing objects according to clusters, which has not been explored in previous work. Each object is randomly assigned membership in three clusters with random μ and predefined $\sigma \in (1.0, 2.0, 4.0)$. Each point is the average of 100 trials over 4 domains. In addition to the performance of OO-POMCP and POMCP, we compare multiple conditions, namely: *Known* where the object locations are known in advanced; *Random* where the agent executes a random policy; *.95/.5* and *.90/1.0* which is OO-POMCP with sensor noise: the first term is sensor accuracy and second term is the standard deviation of a Gaussian noise term for false detection probability. We do not consider discounted cumulative reward because of the large number of actions necessary to find all objects. We can see that OO-POMCP out performs POMCP even with sensor noise. POMCP is also given an additional $1,000$ particles per planning cycle to prevent particle decay. However, POMCP performs near random because of the difficulty of belief tracking in such a large POMDP. OO-POMCP plans with a compactly represented belief in terms of independent distributions which supports robust planning even as the number of objects increases while taking advantage of local spatial-dependencies.

Car Driving Among Pedestrians

Car Driving Among Pedestrian a grid-world domain modeling the behavior of a car agent among walking pedestrians as inspired by [Garg *et al.*, 2019]. The goal of the car agent is to reach to the final destination as soon as possible without colliding with pedestrians. There is positive reward for reaching to the final goal destination $+100$, and negative reward for collisions -100, and -1 for each action to penalize slow movements. The car moves horizontally from left to right in a 5-by-25 map, and can vary its speed by choosing one of three available actions: `STAY`, `ONE-STEP`, and `TWO-STEP`. There are n pedestrians that vertically intersect the path of the car, where the direction and the exact location of each pedestrian is unknown. The belief is defined over each pedestrian location distributed over 3-by-3 cells. Each action yields an observation. The observation function is represented by Gaussian noise, and the accuracy of observation is inversely proportional to the distance between the agent and each pedestrian, like in Rock Sample.

We examine performance as the number of objects (pedestrians) increases while varying the number of simulations. We record the discounted cumulative reward ($\gamma = .95$) of OO-POMCP against POMCP. Each data point is the average of 100 independent trials. This experiment demonstrates how the performance of POMCP degrades as the number of objects increases, reflecting the difficulty of planning with large beliefs. Sequential decision making under OO-POMCP is robust to the number of pedestrians even when considering 15 pedestrians for a small number of simulations.

Conclusion

In this paper, we presented a theoretical analysis and empirical evaluation of the OO-POMCP algorithm. Our paper is the first to make explicit the direct relationship between belief factorization and the number of samples required for online POMDP planning with use of a novel application of Rademacher complexity. We, furthermore, identify a class of

POMDPs that can be tractably solved as the number of objects scale. On a number of domains, we demonstrate the performance of OO-POMCP on this class of POMDPs, while considering a low number of simulations, local spacial dependencies, and large observation spaces. The key insight is modifying the belief representation for efficient tracking.

When modeling a real-world environment, there exists a computational trade off between the number of considered objects and represented object-to-object dependencies. The incorporation of dependencies with a strong induction bias will improve the POMDP solver. However, as shown, naively including all dependencies may harm POMDP planning performance. It is essential to balance tractable planning with the inclusion of dependencies in real-world robotic planners. In future work, we aim to connect our analysis of factorization to concepts of overfitting as in [François-Lavet *et al.*, 2019]. We aim also to explore dynamically factoring the belief on task onset to selectively include useful dependencies, where factorization will be decided via environment interactions in line with recent approaches in robotic interactive perception [Bohg *et al.*, 2017].

Appendix

A. Proof of Theorem 1

For a factored OO-POMDP with $|C|$ object-factors:

$$
\begin{aligned}
Q_\pi(b,a) - \widehat{Q}_\pi(b,a) &= \sum_{i}^{|C|} (Q_\pi(b_i,a) - \widehat{Q}_\pi(b_i,a)) \\
&= \sum_{i}^{|C|} \sum_{c_i} \sum_{r_i,o_i,c_i'} \mathbb{B}\mathbb{P}[r_i + \gamma V_\pi^{a,o_i}(c_i')] - \\
&\qquad \frac{1}{n}\sum_{j}^{n} [r_{i,j} + \gamma V_\pi^{a,o_{i,j}}(c_{i,j}')] \\
&= \sum_{i}^{|C|} \mathbb{E}_{\mathbb{B}}\mathbb{E}_{\mathbb{P}}[f_\pi(r_i,o_i,c_i')|c_i] - \frac{1}{n}\sum_{j=1}^{n} f_\pi((r_i,o_i,c_i')_j) \\
&= \sum_{i}^{|C|} \mathbb{E}_{\mathbb{P}}[f_\pi(r_i,o_i,c_i')] - \frac{1}{n}\sum_{j=1}^{n} f_\pi((r_i,o_i,c_i')_j), \qquad (8)
\end{aligned}
$$

where $\mathbb{B} = p(c_i)$ and $\mathbb{P} = p(r_i, o_i, c_i'|c_i, a)$. Line 4 follows from the law of total expectation. A function in $f_\pi(r_i, o_i, c_i') \in F$ is a possible evaluation of $r + \gamma V_\pi^a(c_i')$ dependent on a policy tree π. $f_\pi(.)$ is within range $[0, \frac{R_{max}}{1-\gamma}]$ by a contraction mapping argument given the reward function is non-negative with a maximum value of R_{max}. We apply the following bound as from [Scott, 2014] for each object-component $c \in C$, $c_i \in c$, and $a \in A$ w.p. $\geq 1 - \delta/(|A|K)$:

$$
\max_{\pi:\bar{B}\to A} \left| \mathbb{E}[f_\pi(r_i,o_i,c_i')] - \frac{1}{n}\sum_{k=1}^{n} f_\pi((r_i,o_i,c_i')_k) \right| \leq
$$

$$
2\widehat{\mathfrak{R}}_{T_{c_i,a}}(F) + \frac{3R_{max}}{1-\gamma}\sqrt{\frac{1}{2n}\log\frac{4|A|K}{\delta}} \qquad (9)
$$

We take the union bound and the maximum empirical Rademacher complexity over $|A|K$ to give the upper bound on 9 to produce Theorem 1.

B. Experimental Details

In this section, we review the experiment details for Figure 2. We evaluated $f_\pi(.)$ in an OO-POMDP domain with 4 objects of unknown location and agent with known location. Each unknown object can exist in 10 possible locations, yielding 10^5 states. The action set A consisted of 6 actions: 2 to move the agent deterministically left or right and 4 to sense if the ith object exists in the current agent location.

The observation and reward functions encode dependencies between objects. A sense action yields observations about the other $n-1$ objects, each with .5 conditional dependency reflecting maximum entropy. The agent receives a reward per state r_s with an additive bonus r_o dependent on the number of objects present per location. Each experiment generates an OO-POMDP with randomized reward and locations of the agent and objects. r_s and r_o was sampled uniformly per location from $[0, 1]$ and $[0, 10 * f(l)]$ respectively; $f(l)$ returns a count of the number of objects present in location l. The actual reward signal has Gaussian noise with standard deviation 0.1.

In Figure 2a, each data point is the average of 20 generated OO-POMDPs, which was sufficient to give low standard error on the estimated empirical Rademacher complexity measurement. The factored model was sampled 20 times per action to give the training set: (c_i, r_i, o_i, c_i') of random factor i. The expectation over Rademacher variables σ was estimated as the average over $1,000$ samples consistent with [Zhu *et al.*, 2009]. We additionally took the average of $1,000$ policy trees for $f_\pi \in F$. This is because the number of policy trees π is doubly exponential in the horizon depth H: $|A|^{|\Omega|H}$ (e.g. a horizon of 4 has approximately 10^7 such policy trees). We report the the maximum measurement over all the training set samples. In Figure 2b and 2c, the factored model was sampled 10 times per action as before. The function $f_\pi(.)$ for each sample was evaluated per policy tree. We reported the average evaluation taken over 20 randomly-generated policy trees.

C. Supplemental Proof

$$
\begin{aligned}
p(o|s',a)p(s'|s,a) &= p(o|s',s,a)p(s'|s,a) \\
&= p(o,s'|s,a)
\end{aligned}
$$

Line 1 and 2 holds by the Markov assumption and Bayes' rule respectively.

References

[Amato and Oliehoek, 2015] Christopher Amato and Frans A Oliehoek. Scalable planning and learning for multiagent pomdps. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Bartlett and Mendelson, 2002] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[Bohg *et al.*, 2017] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[Devin *et al.*, 2018] Coline Devin, Pieter Abbeel, Trevor Darrell, and Sergey Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118. IEEE, 2018.

[Diuk *et al.*, 2008] Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247. ACM, 2008.

[François-Lavet *et al.*, 2019] Vincent François-Lavet, Guillaume Rabusseau, Joelle Pineau, Damien Ernst, and Raphael Fonteneau. On overfitting and asymptotic bias in batch reinforcement learning with partial observability. *Journal of Artificial Intelligence Research*, 65:1–30, 2019.

[Garg *et al.*, 2019] Neha P Garg, David Hsu, and Wee Sun Lee. Despot-α: Online pomdp planning with large state and observation spaces. In *RSS*, 2019.

[Greff *et al.*, 2019] Klaus Greff, Raphaël Lopez Kaufmann, Rishab Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *arXiv preprint arXiv:1903.00450*, 2019.

[Guestrin *et al.*, 2001] Carlos Guestrin, Daphne Koller, and Ronald Parr. Solving factored pomdps with linear value functions. In *Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01) workshop on Planning under Uncertainty and Incomplete Information*, pages 67–75. Citeseer, 2001.

[Halford *et al.*, 1998] Graeme S Halford, William H Wilson, and Steven Phillips. Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences*, 21(6):803–831, 1998.

[Janner *et al.*, 2018] Michael Janner, Sergey Levine, William T Freeman, Joshua B Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-oriented prediction and planning. *arXiv preprint arXiv:1812.10972*, 2018.

[Jiang *et al.*, 2015] Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1181–1189. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[Li *et al.*, 2016] Jue Kun Li, David Hsu, and Wee Sun Lee. Act to see and see to act: Pomdp planning for objects search in clutter. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5701–5707. IEEE, 2016.

[McAllester and Singh, 2013] David A McAllester and Satinder Singh. Approximate planning for factored pomdps using belief state simplification. *arXiv preprint arXiv:1301.6719*, 2013.

[Mitzenmacher and Upfal, 2017] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[Ong *et al.*, 2010] Sylvie CW Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[Pajarinen and Kyrki, 2017] Joni Pajarinen and Ville Kyrki. Robotic manipulation of multiple objects as a pomdp. *Artificial Intelligence*, 247:213–228, 2017.

[Pineau *et al.*, 2006] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Anytime point-based approximations for large pomdps. *Journal of Artificial Intelligence Research*, 27:335–380, 2006.

[Poupart, 2005] Pascal Poupart. *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. Citeseer, 2005.

[Ross *et al.*, 2008] Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-Draa. Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

[Roy *et al.*, 2005] Nicholas Roy, Geoffrey Gordon, and Sebastian Thrun. Finding approximate pomdp solutions through belief compression. *Journal of artificial intelligence research*, 23:1–40, 2005.

[Scott, 2014] Clayton Scott. Lecture notes of eecs 598: Statistical learning theory, January 2014.

[Silver and Veness, 2010] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in neural information processing systems*, pages 2164–2172, 2010.

[Smith and Simmons, 2004] Trey Smith and Reid Simmons. Heuristic search value iteration for pomdps. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 520–527. AUAI Press, 2004.

[Somani *et al.*, 2013] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. In *Advances in neural information processing systems*, pages 1772–1780, 2013.

[Strehl *et al.*, 2007] Alexander L Strehl, Carlos Diuk, and Michael L Littman. Efficient structure learning in factored-state mdps. In *AAAI*, volume 7, pages 645–650, 2007.

[Sunberg and Kochenderfer, 2018] Zachary N Sunberg and Mykel J Kochenderfer. Online algorithms for pomdps with continuous state, action, and observation spaces. In *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Wandzel *et al.*, 2019] Arthur Wandzel, Yoonseon Oh, Fishman Michael, Nishanth Kumar, Lawson Wong L.S., and Stefanie Tellex. Multi-object search using object-oriented pomdps. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[Zhu *et al.*, 2009] Jerry Zhu, Bryan R Gibson, and Timothy T Rogers. Human rademacher complexity. In *Advances in Neural Information Processing Systems*, pages 2322–2330, 2009.