Optimal Continuous State POMDP Planning With Semantic Observations: A Variational Approach

Luke Burks, Ian Loefgren, and Nisar R. Ahmed

***Abstract*—This article develops novel strategies for optimal planning with semantic observations using continuous state partially observable Markov decision processes (CPOMDPs). Two major innovations are presented in relation to Gaussian mixture (GM) CPOMDP policy approximation methods. While existing methods have many desirable theoretical properties, they are unable to efficiently represent and reason over hybrid continuous-discrete probabilistic models. The first major innovation is the derivation of closed-form variational Bayes GM approximations of point-based value iteration Bellman policy backups, using softmax models of continuous-discrete semantic observation probabilities. A key benefit of this approach is that dynamic decision making tasks can be performed with complex non-Gaussian uncertainties, while also exploiting continuous dynamic state-space models (thus avoiding cumbersome and costly discretization). The second major innovation is a new clustering-based technique for mixture condensation that scales well to very large GM policy functions and belief functions. Simulation results for a target search and interception task with semantic observations show that the GM policies resulting from these innovations are more effective than those produced by other state-of-the-art policy approximations, but require significantly less modeling overhead and online runtime cost. Additional results show the robustness of this approach to model errors and scaling to higher dimensions.**

***Index Terms*—AI reasoning methods, Gaussian mixtures (GMs), hybrid systems, partially observable Markov decision processes (POMDPs), sensor fusion, target search and localization.**

I. Introduction

MANY applications of planning under uncertainty require autonomous agents to reason over outcomes in continuous dynamical environments using imprecise but readily available semantic observations. For instance, in search and tracking applications, autonomous robots must efficiently reacquire and localize mobile targets that remain out of view for long periods of time. Planning algorithms must therefore generate vehicle trajectories that optimally exploit "detection" and "no detection" data from onboard sensors [1], [2], as well as semantic natural language observations that can be provided by human supervisors [3]. However, it remains quite challenging to achieve tight optimal integration of vehicle motion planning with nonlinear sensing and non-Gaussian state estimation in large continuous dynamic problem domains.

In recent years, a variety of techniques based on partially observable Markov decision processes (POMDPs) have been developed to address these issues. These include methods that preserve the continuous dynamical nature of the problem through suitable function approximations, rather than discretizing the continuous state space. Of particular interest here are approximations based on Gaussian mixture (GM) models, which flexibly represent complex policy functions and non-Gaussian probability density functions (PDFs) [4], [5]. These techniques enable closed-form manipulation and recursions for producing accurate optimal POMDP policy approximations. However, these state-of-the-art methods suffer from two major drawbacks when dealing with semantic observations. First, they rely on computationally expensive and nonscalable hybrid probabilistic likelihood models for capturing the relationship between discrete semantic sensor data and continuous states. Second, these methods rely on computationally expensive GM condensation techniques to maintain tractability. These issues greatly increase modeling and computational effort, and thus significantly limit the practical applicability and scalability of GM-based POMDP approximations to continuous state decision making problems.

This article presents two technical innovations to directly address these issues. The first novel contribution is an efficient variational Bayes (VB) GM POMDP policy approximation method that allows semantic sensor observations to be accurately yet inexpensively modeled by generalized softmax likelihood models (which otherwise lead to intractable policy and PDF updates for continuous POMDPs). This method is further extended to account for continuous dynamic state-space models. The second novel contribution is the development of a fast and scalable two-stage GM condensation technique for large mixtures. Application of the K-means algorithm to precluster mixands, followed by a Kullback–Leibler divergence (KLD)-based condensation of each cluster and recombination of the resulting mixtures, leads to significantly faster condensation overall with minimal loss of accuracy. This technique is tested across a range of parameters including the dimensionality, initial size, and final size of the mixture. While many different distance and information theoretic divergence measures could be used to partition GM PDFs, the Euclidean distance of mixand means

Manuscript received March 12, 2019; accepted July 25, 2019. This article was recommended for publication by Associate Editor M. Schwager and Editor T. Murphey upon evaluation of the reviewers' comments. This work was supported in part by the Center for Unmanned Aircraft Systems (C-UAS), in part by the National Science Foundation (NSF) Industry/University Cooperative Research Center under NSF Award No. CNS-1650468 along with significant contributions from C-UAS industry members. *(Corresponding author: Nisar Ahmed.)*

The authors are with the Smead Aerospace Engineering Sciences Department, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: luke.burks@colorado.edu; ian.loefgren@colorado.edu; nisar.ahmed@colorado.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2019.2933720

strikes a balance between speed and accuracy that scales well to higher dimensional problems. Simulation results of VB GM policy approximations are provided on a dynamic target search and interception task, showing favorable performance comparisons to other approximation methods as well as robustness to model error. Finally, simulation results for a multirobot localization and planning problem with semantic sensing demonstrate the scalability of the VB policy approximations in higher dimensional settings, where other offline policy approximations are impractical to use.

This article significantly extends the theory and results presented by Bai *et al.* in [6]. Specifically, this article provides a more detailed and generalized derivation of the VB GM policy approximation approach for linear-Gaussian dynamical systems, and provides a more rigorous analysis of the clustering-based GM condensation algorithm. This article also provides additional comparisons to state-of-the-art POMDP policy approximations, assesses the robustness of the VB GM policy approximation to model errors, and shows that the approach scales well to high-dimensional problems. Though outside the scope of this article, the methods developed here have also been adapted and deployed in hardware for collaborative human–robot target search and interception [7].

II. Background and Related Work

A. Motivation and General Problem Description

This article is motivated by dynamic target search and interception tasks. However, the concepts developed here readily generalize to other problems involving decision making under uncertainty in continuous dynamic state spaces, e.g., mobile robot self-localization and motion planning [8]–[10]; inventory control [11]; unmanned aircraft collision avoidance [12]; population dynamics modeling [13]; multitarget radar scheduling [14]; robot arm motion planning and coordinated grasping-based manipulation [15]–[18]; and autonomous driving [16], [19].

The dynamic target search and interception problem considered here consists of a single autonomous mobile robot platform (the seeker), which must seek out, localize, and capture another single mobile entity (the target). The seeker and target dynamics are each described by a finite-dimensional continuous state-space dynamics model. The seeker robot receives a limited set of noisy sensor observations and can use these to make informed decisions about its own movements, which in turn lead to new future observations and possible interception of the target.

It is assumed that the seeker has perfect (or near-perfect) knowledge and observability of its own state, although its actions may result in uncertain state transitions. This can be relaxed to allow for uncertain seeker states, though it is assumed regardless that the seeker states are observable and the search environment is known, such that obstacles and other known hazards are mapped ahead of time. The seeker also has a (possibly imperfect) state-space model of the target, as well as an initial prior belief over target states. The seeker's sensor observations consist of semantic data types that are generated in the continuous space as discrete categorical observations, i.e., positive information in the form of "target detected" and negative information in the form of "no target detected" reports from a visual sensor. Continuous sensor observations may also be present (e.g., relative range and bearing measurements), though the semantic/discrete observations are the distinguishing feature and focus for this article.

Given this setup, the seeker must reason about how to intercept the target in some optimal sense. This article focuses on the problem of safe minimum time capture; that is, the seeker must intercept the target as quickly as possible without colliding with any known obstacles in the environment ("pop-up" hazards and imperfect maps are not considered). Alternative performance measures could be optimized, e.g., maximum probability of capture, minimum mean squared error target localization error, minimum power consumption, etc. Regardless, optimal planning requires the seeker to map the target's state, its own state, and its set of possible actions and observations to the maximization of an overall utility for some planning horizon.

Previous work in controls, data fusion, and robotics has expansively addressed target search and tracking and interception for continuous spaces [2], [20]–[22]. However, optimal planning under uncertainty in continuous spaces with semantic observations remains quite challenging. The hybrid probabilistic nature of this application present challenging data fusion and control problems with highly non-Gaussian uncertainties, which are not present in other approaches that rely on continuous measurements and Gaussian uncertainties. In such a problem, the typical approach is to apply the separation principle, relying on the observability of the state space and properties of the sensor and dynamics models to ensure Gaussian uncertainties over the long term. However, the separation principle is not guaranteed to produce optimal results for planning under uncertainty with semantic observations, since these are typically nonlinear and can lead to highly non-Gaussian uncertainties. Fusion approaches applied to this problem must be able to accommodate arbitrary uncertainties using non-Gaussian sensor and dynamics models.

B. Semantic Sensing and Data Fusion

Sensors typically provide continuous numeric observations, such as a range or bearing measurement. However, some sensors provide categorical observations, e.g., the output of a visual object detection algorithm that reports when an object is in a camera sensor's field of view, or a human-generated report that a target is west of a landmark. Such semantic observations map to discrete regions in a continuous space, where the regions are not necessarily exclusive. For example, placing a target at the edge of a camera's view could generate either a positive (true detection) or negative (false miss) observation from the identification algorithm, with some probability for each outcome. When these probabilities are cast in a likelihood model, they provide useful negative information in Bayesian reasoning for target tracking [23] [24], as well as probabilistic generative models of semantic observations for planning problems [25].

In collaborative human–robot search problems, nonrobot sensors such as surveillance cameras, unattended ground sensors, or human teammates who generate natural language data can be modeled as semantic data sources [3], [26], [27]. The seeker robot can then augment its decision making process by incorporating actions to actively point and poll these sensors in a closed-loop manner, resulting in a fully integrated hybrid sensing and planning problem. Kurniawati *et al.* [28], [29] also considered this problem from the standpoint of myopic value of information (VOI) reasoning to determine whether querying a particular sensor will result in a better decision in the long run (i.e., improved utility), despite the cost of using the sensor and regardless of the sensor's observation. These approaches require online optimization and inference for decision making within a probabilistic graphical model, and hence decouple the planning and sensing problems to ensure computational tractability. The approach described in this article can be used to solve for combined motion planning and human querying policies offline, thus avoiding high computational cost and achieving tighter integration of planning and sensing with complex uncertainties. Application to the full semantic active sensing problem with human–robot teams is not treated here, but has been implemented and examined in related work [7].

Semantic data fusion has major consequences for online state estimation and representation. The negative information carried by such data can change the state belief in highly nonlinear ways via the "scattering effect" [21]. This temporarily increases the differential entropy of a continuous target state PDF by introducing non-Gaussian features like holes, skewness, and multiple peaks. Yet, many data fusion and estimation approaches rely heavily on Gaussian state PDFs and likelihood models; these can lose significant information relative to the true target state distribution and, thus, lead to suboptimal closed-loop search/localization policies. Extensions of these methods generally rely on approximations of PDFs and observation likelihoods via normalized and unnormalized GM functions, respectively [4], [30]. These methods exploit the fact that, for recursive Bayesian updates, the product of GM state prior distributions and GM semantic likelihood functions is always guaranteed to be another GM. It is fairly well-established that normalized GMs provide highly flexible models for non-Gaussian state estimation, especially if GM condensation techniques are applied to control mixand growth across successive mixture operations [31], [32]. However, the number of parameters required for unnormalized GMs to model semantic data likelihoods in two or more continuous state dimensions scales quite poorly and quickly becomes computationally impractical for optimal planning. Previous work also showed that semantic observations could be modeled via softmax functions and fused into (normalized) GM PDFs for recursive Bayesian state estimation [3], [27]. This concept is significantly extended here for updating unnormalized GM policy functions, which also accounts for the tight coupling between optimal sensing and planning.

C. Continuous State Space Planning Under Uncertainty

Dynamic target search and interception problems feature many types of stochastic uncertainty, including dynamic process noise and sensor errors. Planners based on POMDPs are well-suited to handle such uncertainties. POMDPs can theoretically support arbitrary dynamics, state beliefs (probability distributions), and sensor models, and thus encode a broad range of general optimal decision making problems when specified with an appropriate reward function. POMDP policies can operate on arbitrary target state PDFs, as long as Bayesian belief updates to the target state PDF are carried out. POMDPs also naturally account for VOI when integrated with sensor tasking and information gathering actions. In practice, however, POMDPs can be unworkable due to the curse of dimensionality in discrete spaces and problems with tractability and representation in continuous spaces. The key challenge is the need to solve a Markov decision process (MDP) over the state belief space to obtain optimal decision making policies. This is impractical to do exactly for all but the most trivial problems [33]. Hence, it is generally necessary to resort to approximate solutions.

Discrete space POMDP approximations have been regularly applied to target search and interception in prior work; benchmark applications include "tag/avoid" [34], [35] and laser tag [36]. These approximations generate offline policies for target interception based on a discretization of the continuous state space. However, solving the belief space MDP for these problems carries the curse of dimensionality. For a problem with N discrete states, policies must be found over the continuous space of all N-dimensional probability distributions, and thus become intractable to represent. Approximations based on point-based value iteration (PBVI) [34] attempt to solve the POMDP at specific "tentpole" beliefs, allowing the policy to be interpolated at other beliefs [35], [37]. Other methods attempt to compress the belief space onto a lower dimensional manifold [38], approximate the POMDP as a single step MDP with observations, e.g., Q-MDP [39], or use sample states to build trees of potential histories [40] or scenarios [41] in an online fashion during runtime. One of these online sampling methods, partially observable Monte Carlo planning (POMCP) [25], has also been adapted recently to continuous state spaces [42].

Recent years have seen the development of several POMDP policy approximations for continuous state, action, and observation spaces. These include a variety of belief representations, and address the combination of continuous states with discrete or continuous actions and observations.

Several continuous POMDP approximation approaches rely on sampling methods [12], [43], in a similar or extended version of the discrete space sampling approaches. Local policy approximations for continuous observations have been applied using Gaussian state beliefs [15], [44]. Belief space roadmap techniques [45] have had success in computing policies as paths directly in belief space for linear Gaussian systems with continuous observations. Also related are policy approximations inspired by linear-quadratic-Gaussian optimal control for motion planning under uncertainty [10], [46].

One family of continuous POMDP approximations extends the PBVI discrete approach to continuous spaces [5]. This approach uses GMs to approximate arbitrarily complex PDF beliefs, state transitions, and observation likelihoods of the POMDP. Beliefs are updated via the Gaussian Sum filter, exploiting the fact that GMs become universal function approximators as the number of mixing components (mixands) becomes large. Bai *et al.* [4] and [47] extended this idea to address hybrid dynamical systems, where the state transition model switches in different parts of the state space.

These existing techniques are generally ill-suited for complex planning problems in continuous state spaces with semantic observations, such as the target search and interception task considered here. Discretization-based methods lead to an undesirable tradeoff between state space size and fidelity of system dynamics, with larger spaces requiring coarse discretizations that fail to capture subtleties in the target model. Online approximations require significant tuning effort and incur high computational costs during execution, and even state-of-the-art approaches such as POMCP [25] can suffer from suboptimal worst case behavior and incur high regret [48]. Continuous state policy approximations such as [4] and [12] have either relied on the assumption of continuous observations (and are thus not amenable to semantic observations), or constructed semantic observation likelihoods out of GMs. In the latter case, such models are chosen to facilitate calculations and maintain closed-form recursions, but drive up computation cost of policy searches and scale poorly with state dimension N due to the number of mixands required to accurately specify likelihoods. This article develops a more scalable alternative likelihood representation using softmax models for PBVI type policy solutions with GM belief representations. Existing GM-based policy approximations for semantic observations also make the simplifying assumption that state transitions are independent of the current state. This limits their applicability to dynamic search and tracking problems, where target dynamics are often described by flexible linear time-invariant (LTI) models. The research developed here relaxes this assumption and, thus, generalizes. Finally, unlike Monte Carlo-based approximations [9] or learning-based belief compression methods [38], the approach developed here provides deterministic policy approximations for a given set of “tentpole” beliefs.

D. POMDP Preliminaries

Formally, a POMDP is described by the 7-tuple $(S, A, T, R, \Omega, O, \gamma)$, where: S is a set of states; A is a set of $|A|$ discrete actions a; T is a discrete time probabilistic transition mapping from state s_t at time t to state s_{t+1} at time $t+1$ given some a; R is the immediate reward mapping over (s, a) pairs; Ω is a set of observations o with $N_o = |\Omega|$ possible outcomes; O is the likelihood mapping from states to observations; and $\gamma \in [0, 1]$ is a discount factor. An agent whose decision making process is modeled by a POMDP seeks to maximize a utility function defined by the expected future discounted reward: $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$, where $s_t \in S$ is the state at discrete time t, and $a_t \in A$. The expectation operator $\mathbb{E}[\cdot]$ reflects that the agent lacks full knowledge of s_t. It must instead rely on the noisy process model T and observation model O to update a Bayesian belief function $b(s_t) = p(s_t|a_{1:t}, o_{1:t})$, which summarizes all available information for reasoning about present and possible future states. An optimal decision making policy $\pi(b(s_t)) \to a_t$ must therefore be found for any possible belief $b(s_t)$. Since POMDPs are equivalent to MDPs over state beliefs $b(s_t)$, exact policies are impossible to compute for all but the simplest problems.

Let $s, s' \in S$ be arbitrary states to which any (approximate) policy must apply such that $s \mapsto s'$ via T (i.e., for any $t \to t+1$). One well-known family of techniques for computing approximate POMDP policies offline is PBVI [34]. These methods approximate π at a finite set of beliefs $\mathcal{B}_0 = \{b_1(s), \ldots, b_{N_B}(s)\}$, for which explicit finite-horizon Bellman equation recursions can be performed to obtain locally optimal actions in the neighborhood of each $b_i(s), i = 1, \ldots, N_B$. When S is a set of discrete states with N possible outcomes, then $b(s) \in \mathbb{R}^N$ such that $\sum_{s=1}^{N} b(s) = 1$. In this case, PBVI policies are represented by a set Γ of N_α vectors $\alpha \in \mathbb{R}^N$. The α vectors mathematically represent hyperplanes that encode value functions for taking particular actions at a given belief. The action a recommended by the policy for a given $b(s) \in \mathbb{R}^N$ is found as the action associated with $\arg\max_{\alpha \in \Gamma} < \alpha, b(s) >$, where $< \cdot >$ is the inner product. A number of methods exist for generating typical sample beliefs, e.g., starting with a large set of $b_i(s)$ sampled from the reachable belief space by random simulation [34] (as in this article), or propagating a small initial belief set in between recursive Bellman updates for α vector computations to approximate optimal reachable belief sets [35].

When s is a continuous random vector such that $s \in \mathbb{R}^N$ with support $\mathcal{S}(S)$, it is natural to represent $b(s)$ as a PDF, where $\int_{\mathcal{S}(S)} b(s) ds = 1$. In such cases, continuous state POMDPs (CPOMDPs) can be formulated by specifying T, R, O and $\alpha(s)$ as suitable continuous functions over s. Although $b(s)$ can sometimes be represented by simple parametric models such as Gaussian PDFs [49], $b(s)$ is in general analytically intractable for arbitrary T and O models that represent nonlinear/non-Gaussian dynamics and semantic sensor observations. Therefore, $b(s)$ must also be approximated to derive a suitable set Γ of $\alpha(s)$, such that the (approximate) optimal PBVI policy $\pi(b(s))$ is defined by the action associated with $\arg\max\langle\alpha(s), b(s)\rangle$.

E. Gaussian Mixture CPOMDPs

Finite GM models provide a general and flexible way to approximate arbitrary functions $f(s)$ of interest for CPOMDPs, where

$$f(s) = \sum_{m=1}^{G} w_m \phi(s|\mu_m, \Sigma_m) \tag{1}$$

is a GM defined by G weights $w_m \in \mathbb{R}_{0+}$, means $\mu_m \in \mathbb{R}^N$, and symmetric positive semidefinite covariances $\Sigma_m \in \mathbb{R}^{N \times N}$ for the multivariate normal component PDF (“mixand”) $\phi(s|\mu_m, \Sigma_m)$, such that $\sum_{m=1}^{G} w_m = 1$ to ensure normalization when $f(\cdot)$ represents a PDF (this condition need not apply otherwise). Bai *et al.* [5] showed the following: let A describe a discrete action space with finite realizations a, $T = p(s_{t+1}|s_t, a) = p(s'|s, a)$ a Gaussian state transition PDF, $O = p(o_{t+1}|s_{t+1}) = p(o'|s')$ a GM observation likelihood, and $R(s_t, a_t = a) = r_a(s_t) = r_a(s)$ a continuous GM reward function for each a, such that

$$T = p(s'|s, a) = \phi(s_{t+1}|s_t + \Delta(a), \Sigma^a) \tag{2}$$

$$O = p(o'|s') = \sum_{l=1}^{M_o} w_l \phi(s_{t+1}|\mu_l, \Sigma_l) \tag{3}$$

$$R = r_a(s) = \sum_{u=1}^{M_r} w_u \phi_u(s_t|\mu_u^a, \Sigma_u^a) \tag{4}$$

(where $\phi(\mu, \Sigma)$ is a Gaussian with mean μ and covariance matrix Σ, and $\Delta(a)$ is the change in $s = s_t$ due to action a); then PBVI approximations to $\pi(b(s))$ can be found from closed-form GM Bellman recursions for a finite set of GM functions $\alpha(s_t)$ defined over some initial set of GM beliefs $b(s)$. Note that $r_a(s_t)$ is generally an unnormalized GM, with possibly negative mixture weights such that $\int_{\mathcal{S}(s_t)} r_a(s_t) ds_t \neq 1$. This allows the CPOMDP to penalize certain configurations of continuous states with discrete actions and thus discourage undesirable agent behaviors. However, T must obey the usual constraints for PDFs, such that $\int_{\mathcal{S}(s_{t+1})} p(s_{t+1}|s_t, a) ds_{t+1} = 1$. The observation likelihood must also obey $\sum_{o_{t+1}} p(o_{t+1}|s_{t+1}) = 1$ for all s_{t+1}, where o_{t+1} is a discrete random variable describing a semantic observation. As such, $p(o_{t+1}|s_{t+1})$ can be a GM with strictly positive weights but unnormalized weights (i.e., which do not sum to 1) to model the conditional probability for o_{t+1}'s outcomes as a continuous function of s_{t+1}.

If the belief PDF $b(s)$ is a J-component GM

$$b(s) = \sum_{q=1}^{J} w_q \phi(s|\mu_q, \Sigma_q) \tag{5}$$

then it is possible to arrive at a finite set $\Gamma_n = \{\alpha_n^1, \alpha_n^2, \ldots, \alpha_n^{N_\alpha}\}$ of $\alpha(s)$ functions for an n-step look-ahead decision starting from $b(s)$, such that

$$\alpha_n^i(s) = \sum_{k=1}^{M} w_k^i \phi(s|\mu_k^i, \Sigma_k^i) \ \ (\alpha_n^i \in \Gamma_n) \tag{6}$$

and the optimal value function $V_n^*(b(s))$ is approximately

$$V^*(b(s)) \approx \max_{\alpha_n^i} < \alpha_n^i, b(s) > \tag{7}$$

$$\begin{aligned} &< \alpha_n^i, b(s) > \\ &= \int_{\mathcal{S}(S)} \left[\sum_{k=1}^{M} w_k^i \phi(s|\mu_k^i, \Sigma_k^i)\right] \left[\sum_{q=1}^{J} w_q \phi(s|\mu_q, \Sigma_q)\right] ds \end{aligned} \tag{8}$$

$$= \sum_{k,q}^{M\times J} w_k^i w_q \phi(\mu_q|\mu_k^i, \Sigma_q + \Sigma_k^i) \int_{\mathcal{S}(S)} \phi(s|c_1, c_2) ds \tag{9}$$

$$= \sum_{k,q}^{M\times J} w_k^i w_q \phi(\mu_q|\mu_k^i, \Sigma_q + \Sigma_k^i) \tag{10}$$

$$c_2 = [(\Sigma_k^i)^{-1} + (\Sigma_q)^{-1}]^{-1}$$

$$c_1 = c_2[(\Sigma_k^i)^{-1}\mu_k^i + (\Sigma_r)^{-1}\mu_q]$$

which follows from the fact that the product of two Gaussian functions is another Gaussian function. The n-step look-ahead approximation is commonly in PBVI methods, where n is large enough such that V_n^* does not change appreciably and, thus, converges closely to the infinite horizon V^*.

The $\alpha_n^i \in \Gamma_n$ functions are computed using n-step policy rollouts, starting from N_B different initial GM beliefs $\mathcal{B}_0 = \{b_1(s), \ldots, b_{N_B}(s)\}$. In each step, for each $b \in \mathcal{B}_0$, each α_{n-1}^i function's value is updated via the "Bellman backup" equations, which perform point-wise value iteration to capture the effects of all possible observations and actions on the accumulated expected reward for future time steps $0, \ldots, n$. These lead to the recursions

$$\alpha_{a,j}^i(s) = \int_{s'} \alpha_{n-1}^i(s') p(o' = j|s') p(s'|s, a) ds' \tag{11}$$

$$\alpha_n^i(s) = r_a(s) + \gamma \sum_{j=1}^{N_o} \arg\max_{\alpha_{a,j}^i} (< \alpha_{a,j}^i, b(s) >) \tag{12}$$

where $\alpha_{a,j}^i(s)$ is an intermediate function corresponding to a value for a given action-observation pair (a, j) at step n, and $\alpha_n^i(s)$ is the discounted marginalization over all observations of the intermediate function that maximizes the belief being backed up, summed with the reward function. The action then associated with each α_n^i is the one, which maximized the value marginalized over observations. Since $p(s_{t+1}|s_t, a)$ is Gaussian, $p(o_{t+1}|s_{t+1})$ is a GM, and $r_a(s_t)$ is a GM for each action a, the Bellman backups yield closed-form GM functions for $\alpha_n^i(s_t)$. Since the GM function for $r_a(s_t)$ can have negative weights and values, it follows that each GM function $\alpha_n^i(s_t)$ can also take on negative weights and values.

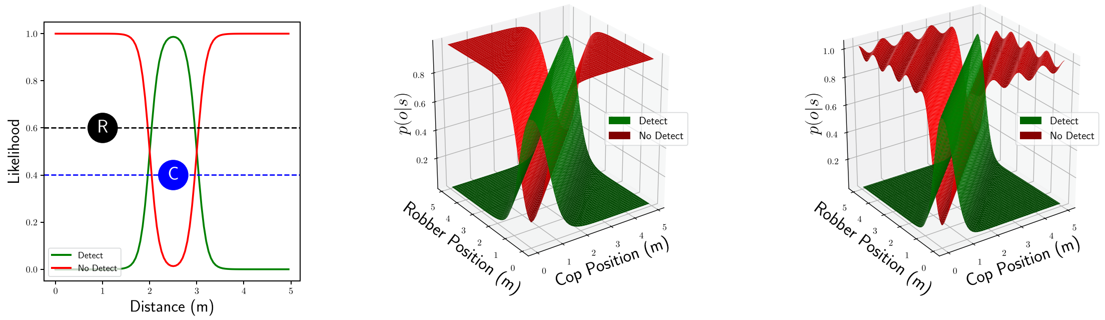
This GM formulation scales well for continuous state spaces where $N \geq 2$, and naturally handles highly non-Gaussian beliefs $b(s)$ stemming from nonlinear/non-Gaussian state process and observation models in a deterministic manner. In contrast to approximations that discretize S to transform the CPOMDP into a standard discrete state POMDP (and thus scale badly for large N), the complexity of the CPOMDP policy (i.e., the required number of mixture terms for each $\alpha_n^i(s)$) depends only on the complexity of the dynamics for the state belief PDF $b(s)$, rather than the number of continuous states N. Furthermore, since the Bellman backup equations can be performed entirely offline using a set of initial beliefs $\mathcal{B}_0$, the resulting policy induced by the final set of $\alpha_n^i(s)$ functions can be quickly and easily computed online: as the agent obtains new state beliefs $b(s_t) \rightarrow b(s_{t+1})$ over time via the standard Bayes' filter equations (which yield the general Gauss sum filter in this case)

$$b(s_{t+1}) \propto p(o_{t+1}|s_{t+1}) \int_{S(s_t)} p(s_{t+1}|s_t, a) b(s_t) ds_t \tag{13}$$

the optimal action a to take for $b(s_{t+1})$ is the one associated with the $\alpha_n^i \in \Gamma_n$ satisfying $\arg\max_{\alpha_n^i} < (\alpha_n^i), b(s_{t+1}) >$.

F. Limitations for Hybrid Continuous-Discrete Reasoning

The likelihood model $O = p(o_t|s_t)$ must describe a valid hybrid (continuous-discrete) probability distribution, such that $\sum_{o_t} p(o_t|s_t) = 1 \ \forall s \in \mathcal{S}(S)$. For each possible outcome o_t, this is typically modeled by an unnormalized GM [5],



(a) (b) (c)

Fig. 1. Model for two colinear robots, one a cop and the other a robber. The cop's robber detection observation likelihood can be modeled as a 9 parameter MMS model (b), or a 624 parameter GM model (c). (a) Colinear model. (b) Softmax likelihood model. (c) GM No Detect.

$p(o_t|s_t) \approx \sum_{l_o=1}^{L_o} w_o \phi(s_t|\mu_{l_o}, \Sigma_{l_o})$, such that $\sum_{o_t} p(o_t|s_t) \approx 1$ everywhere. Such models preserve the closed-form Bellman updates required for PBVI, but are difficult and labor intensive to specify. In particular, for state dimension $N \geq 2$, L_o must be very large for each possible o_t outcome to ensure that the normalization requirement is satisfied for all s_t and that the desired probabilities $p(o_t|s_t)$ are modeled accurately. This effectively turns $p(o_t|s_t)$ into a "soft grid" discretization and severely restricts the scalability of GM policy approximation.

Another issue is the fact that the number of multiplications and summations in each step of the Bellman recursions (11), (12) grows with the number of resulting GM components from J mixands in $b(s)$ and K mixands in $\alpha_n^i(s)$ to $K \times J$ mixands. The number of terms in $\alpha_n^i(s)$ also grows with each summation over N_o observations. GM condensation methods are thus needed to control the size of the policy functions $\alpha_n^i(s)$ between backup steps for offline policy approximation, as well as the size of the state belief GM PDF $b(s)$ between Bayes' filter updates for online policy evaluation. Bai *et al.* [4], [5] proposed different general methods for condensing GM functions in CPOMDP policy approximations, although in principle any number of GM merging algorithms developed in the target tracking and data fusion literature could also be applied [31], [50]. However, for large-scale problems such as dynamic target search and tracking, it is not uncommon for offline Bellman backups and online policy evaluations to rapidly produce hundreds or even thousands of new mixands in just one backup step or Bayes' filter prediction/measurement update. As discussed in Section III-B, existing GM merging methods tend to be computationally expensive and slow for such large mixtures. This issue is exacerbated by use of dense unnormalized GM models for $p(o_t|s_t)$, which introduce additional policy approximation errors if normalization is not guaranteed for all $s_t \in \mathcal{S}(S)$. Offline policy approximation and online policy evaluation thus become expensive.

Existing GM-based CPOMDP approximation methods also use simplified random walk state transition models $p(s_{t+1}|s_t, a)$ where s_{t+1} is modeled as the result of shifting s_t by a specific distance in $\mathcal{S}(S)$ for a given action a plus some additional random noise component. It is thus not obvious how these methods can be used with more sophisticated state dynamics models, e.g., kinematic linear nearly constant velocity (NCV) models commonly used in dynamic target tracking [51], where the position components of s_{t+1} depend on velocity components in s_t. This severely limits the applications for GM-based CPOMDP approximations.

G. *Target Search Example With Semantic Observations*

Consider a two-state CPOMDP in which an autonomous robot "cop" attempts to localize and catch a mobile "robber," where each moves along a single dimension [see Fig. 1(a)]. Here, $S = \mathbb{R} \times \mathbb{R}$ consists of $N = 2$ bounded continuous random variables at each discrete time t, $s_t = [\text{Cop}_t, \text{Rob}_t]^T$, $\text{Cop}_t \in (0, 5)$, $\text{Rob}_t \in (0, 5)$, where Cop and Rob denote the state variable for the respective agents. The robber executes a random walk

$$p(\text{Rob}_{t+1}) = \phi(\text{Rob}_{t+1}|\text{Rob}_t, 0.5). \tag{14}$$

The cop chooses a movement direction from among three noisy actions $A \in \{\text{left}, \text{right}, \text{stay}\}$, such that

$$p(\text{Cop}_{t+1}|\text{Cop}_t, \text{left}) = \phi(\text{Cop}_{t+1}|\text{Cop}_t - 0.5, 0.01) \tag{15}$$

$$p(\text{Cop}_{t+1}|\text{Cop}_t, \text{right}) = \phi(\text{Cop}_{t+1}|\text{Cop}_t + 0.5, 0.01) \tag{16}$$

$$p(\text{Cop}_{t+1}|\text{Cop}_t, \text{stay}) = \delta(\text{Cop}_{t+1}, \text{Cop}_t). \tag{17}$$

The cop is rewarded for remaining within a set distance of the robber's position, and penalized otherwise

$$r(|\text{Rob}_t - \text{Cop}_t| \leq 0.5) = 3 \tag{18}$$

$$r(|\text{Rob}_t - \text{Cop}_t| > 0.5) = -1. \tag{19}$$

The cop receives binary semantic observations from an onboard visual detector, $o_t \in \{\text{"robber detected"}, \text{"robber not detected"}\}$. Fig. 1(b) and (c) shows unnormalized GM models for the semantic "detection" and "no detection" likelihoods, which are respectively parameterized by 8 and 200 isotropic Gaussian components. These models follow the specification of $O = p(o_t|s_t)$ suggested by Bai *et al.* [4], [5], and require 624 parameters total. Since it is expected that the cop will gather mostly "no detection" observations in a typical scenario, it is clear from (11) that the number of mixing components for $\alpha_{a,o}^i(s)$ will grow by a factor of at least 600 on a majority of the intermediate Bellman backup steps for offline policy approximation. Likewise, (13) implies that the number of mixture components for $b(s_{t+1})$ will grow by a factor of at least 600 on each update of the Bayes' filter whenever the target is not detected. This example shows that, even for relatively small problems, unnormalized GM likelihood models are inconvenient for approximating or evaluating optimal policies in continuous state spaces.

III. Variational CPOMDP Policy Approximation

As discussed in [3] and mentioned in [4], semantic observation likelihoods are ideally modeled by self-normalizing functions like the softmax model

$$p(o_t = j|s_t) = \frac{\exp(w_j^T s_t + b_j)}{\sum_{c=1}^{N_o} \exp(w_c^T s_t + b_c)} \tag{20}$$

where $w_1, \ldots, w_{N_o} \in \mathbb{R}^N$ and $b_1, \ldots, b_{N_o}$ are the vector weight parameters and scalar bias parameters for each categorical outcome of o_t given s_t. In addition to ensuring $\sum_j p(o_t = j|s_t) = 1 \; \forall s_t \in \mathcal{S}(S)$, softmax functions require relatively few parameters compared to GM likelihoods, and scale well to higher dimensional spaces. Fig. 1(b) shows how the cop's semantic observation likelihood can be easily modeled with a softmax function featuring three semantic categorical classes (two of which collectively represent the "no detect" observation via the generalized "multimodal softmax" (MMS) formulation [27]). Unlike the GM likelihood function approximation in Fig. 1(c), the softmax model only requires nine parameters.

In general, softmax parameters are easily synthesized to conform to *a priori* sensing geometry information and quickly calibrated with training data [27], [52]. Building on previous work, this assumes linear boundaries between softmax classes. However, since the product of Gaussian and softmax functions is analytically irreducible, modeling $p(o_t|s_t)$ via softmax functions breaks the recursive nature of the α function updates for GM-based PBVI approximations. This section addresses this issue in a novel way using a VB inference approximation. The VB approximation allows the product of each Gaussian term within a GM and a softmax likelihood function to be approximated as a GM. This restores the closed-form Bellman recursions for GM α approximations while keeping the resulting number of mixands in the result to a minimum. This VB approximation is inspired by a very similar technique developed in [3] to approximate (13) for the Bayesian filtering problem, when $b(s_{t+1})$ is a GM PDF and $p(o_{t+1}|s_{t+1})$ is a softmax model. Here, the approximate VB inference technique is generalized to the dual problems of Bayesian filtering and optimal action selection under uncertainty for CPOMDPs. This technique is then extended for nonrandom walk transition functions to permit use of LTI state-space models in the GM-based Bellman recursions.

A. Variational PBVI for Softmax Semantic Likelihoods

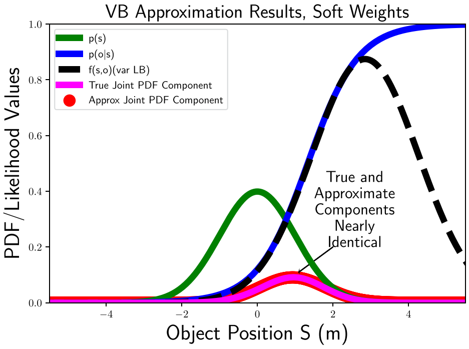


Fig. 2. 1-D VB approximation example: approximate PDF-likelihood product (red) aligns closely with true product (pink).

To use softmax $p(o'|s')$ functions in the GM-based PBVI CPOMDP policy approximation described in (11) and (12), the local VB approximation for hybrid inference with softmax models developed in [3] is used to approximate the product of a softmax model and a GM as a variational GM

$$\begin{aligned}\alpha_{n-1}^i(s')\, p(o' = j|s') &= \left[\sum_{k=1}^{M} w_k^i \phi(s'|\mu_k^i, \Sigma_k^i)\right] \left[\frac{\exp(w_j^T s' + b_j)}{\sum_{c=1}^{N_o} \exp(w_c^T s' + b_c)}\right] \\ &\approx \sum_{h=1}^{M} w_h \phi(s'|\mu_h, \Sigma_h).\end{aligned} \tag{21}$$

Fig. 2 shows the key idea behind this VB approximation using a toy one-dimensional (1-D) problem. The softmax function [blue curve, e.g., representing $p(o'|s')$ in (21)] is approximated by a lower bounding variational Gaussian function (black curve). The variational Gaussian is optimized to ensure the product with another Gaussian function (green, e.g., representing a single mixand of α_{n-1}^i) results in a good Gaussian approximation (red dots) to the true non-Gaussian (but unimodal) product of the original softmax function and Gaussian functions (solid magenta). More formally, the VB update derived in [3] for approximating the product of a normalized Gaussian (mixture) PDF $p(s')$ and a softmax function $p(o'|s')$ can be adapted and generalized to approximate the product of an *unnormalized* Gaussian (mixture) $\alpha_{n-1}^i(s')$ (from the intermediate Bellman backup steps) and softmax likelihood. In the first case, consider the posterior Bayesian PDF for a Gaussian prior $p(s')$ given $o' = j$

$$p(s'|o') = \frac{p(s')p(o'|s')}{p(o')} = \frac{1}{C}\phi(s'|\mu, \Sigma)\frac{\exp(w_j^T s' + b_j)}{\sum_{c=1}^{N_o} \exp(w_c^T s' + b_c)}$$

$$C = \int_{-\infty}^{\infty} \phi(s'|\mu, \Sigma)\frac{\exp(w_j^T s' + b_j)}{\sum_{c=1}^{N_o} \exp(w_c^T s' + b_c)} ds'$$

By approximating the softmax likelihood function as an unnormalized variational Gaussian function $f(o', s')$, the joint PDF and normalization constant C can be approximated as

$$p(s', o') \approx \hat{p}(s', o') = p(s')f(o', s')$$

$$C \approx \hat{C} = \int_{-\infty}^{\infty} \hat{p}(s', o') ds'.$$

The key trick here is that, for any $j \in \Omega$, it is possible to ensure $f(o' = j, s') \leq p(o' = j|s')$ by construction [3], using the variational parameters y_c, γ, and ξ_c such that

$$f(o'=j,s') = \exp\left\{g_j + h_j^T s' - \frac{1}{2}s'^T K_j s'\right\} \tag{22}$$

$$g_j = \frac{1}{2}\left[b_j - \sum_{c\neq j} b_c\right] + \gamma\left(\frac{N_o}{2} - 1\right) + \sum_{c=1}^{N_o} \frac{\xi_c}{2} + \lambda(\xi_c)[\xi_c^2 - (b_c - \alpha)^2] - \log(1 + \exp\{\xi_c\}) \tag{23}$$

$$h_j = \frac{1}{2}\left[w_j - \sum_{c\neq j} w_c\right] + 2\sum_{c=1}^{N_o} \lambda(\xi_c)(\alpha - b_c)w_c \tag{24}$$

$$K_j = 2\sum_{c=1}^{N_o} \lambda(\xi_c) w_c w_c^T. \tag{25}$$

Since $f(o'=j,s') \leq p(o'=j|s')$ for any choice of the variational parameters, it follows that $\hat{C} \leq C$. As such, the variational parameters, which produce the tightest lower bound $\hat{C}$ can be found through an iterative expectation-maximization algorithm, which requires alternately reestimating $\hat{p}(s'|o')$ given new values of the variational parameters, and then recomputing the variational parameters based on new expected values of s' from $\hat{p}(s'|o')$. Upon convergence of $\hat{C}$ to a global maximum, the product $p(s',o'=j) = p(s')p(o'=j|s')$ becomes well-approximated by the product $\hat{p}(s',o'=j) = p(s')f(o'=j|s')$, which is another (unnormalized) Gaussian

$$\hat{p}(s',o'=j) = \exp\left\{(g_p + g_j) + (h_p + h_j)s' - \frac{1}{2}s'^T(K_p + K_j)s'\right\}. \tag{26}$$

Normalizing this joint distribution by $\hat{C}$ gives the posterior Gaussian PDF approximation $\hat{p}(s'|o') = \phi(s'|\mu_h, \Sigma_h)$. The approximation of the product of a GM PDF with a softmax model follows immediately from fact that this product is a sum of weighted products of individual Gaussians with the softmax model, where each individual product term can be approximated via VB.

This approximation is now adapted to the case where the "prior" GM PDF over s' is an unnormalized GM function $\alpha_{n-1}^i(s')$. The results from the above derivation must simply be multiplied by the normalizing constant $\hat{C}$ to obtain the approximate joint $\hat{p}(s',o'=j)$ for each mixture term instead. This allows (11) for the intermediate α function update in the PBVI backup to be approximated as

$$\alpha_{a,j}^i(s) = \int_{s'} \alpha_{n-1}^i(s')p(o'=j|s')p(s'|s,a)ds' \tag{27}$$

$$= \int_{s'} \left[\sum_{k=1}^{M} w_k^i \phi(s'|\mu_k^i, \Sigma_k^i)\right]\left[\frac{\exp(w_j^T s' + b_j)}{\sum_{c=1}^{N_o}\exp(w_c^T s' + b_c)}\right] \times [\phi(s'|s + \Delta(a), \Sigma^a)]\, ds' \tag{28}$$

Algorithm 1: VB-POMDP Backup.

```
Input: $b \in B_0, \Gamma_{n-1}$
for each $\alpha_{n-1} \in \Gamma_{n-1}, a \in A, j \in \Omega$ do
    $\alpha_{a,j}(s) \leftarrow \sum_h w_h \phi(s|\mu_h - \Delta(a), \Sigma^a + \Sigma_h)$
    $\alpha_n(s) = r_a(s) + \gamma \sum_j \arg\max_{\alpha_{a,j}}(< \alpha_{a,j}, b >)$
end for
return: $\alpha_n(s)$
```

$$\approx \int_{s'} \sum_{h=1}^{M} w_h \phi(s|\hat{\mu}_h, \hat{\Sigma}_h)\phi(s'|s + \Delta(a), \Sigma^a)ds' \tag{29}$$

$$\rightarrow \alpha_{a,j}^i(s) \approx \sum_{h=1}^{M} w_h \phi(s|\hat{\mu}_h - \Delta(a), \hat{\Sigma}_h + \Sigma^a) \tag{30}$$

where $(\Delta(a), \Sigma^a)$ are known constants for each a. In practice, the intermediate $\alpha_{a,j}(s)$ functions do not depend on the belief being backed up, and can therefore be calculated once per iteration over all beliefs. Algorithm 1 summarizes how the GM-based PBVI updates developed in Section II are thus modified to use the VB approximation for softmax semantic observation likelihoods. The VB algorithm does introduce some approximation error. The possibility of using the bound on the log-likelihood to estimate this error across backups will be explored in future work.

Following [3], recursive semantic observation updates to GM $b(s_{t+1})$ PDFs can also be carried out online during execution of these policies using softmax likelihoods with the VB approximation, as shown in Fig. 3

$$b(s_{t+1}) \propto p(o_{t+1} = j|s_{t+1}) \int_{s_t} p(s_{t+1}|s_t, a)b(s_t)ds_t \tag{31}$$

$$= \left[\sum_{q=1}^{J} w_q \phi(s_{t+1}|\mu_q + \Delta(a), \Sigma^a + \Sigma_q)\right] \times \left[\frac{\exp(w_j^T s_{t+1} + b_j)}{\sum_{c=1}^{N_o}\exp(w_c^T s_{t+1} + b_c)}\right] \approx \sum_{q=1}^{J} w_q \phi(s_{t+1}|\mu_q, \Sigma_q). \tag{32}$$

In this example, the resulting posterior GM PDF for the "no detection" update has only four components,[1] thus demonstrating that parametrically simpler softmax models drastically reduce the complexity of inference compared to unnormalized GM likelihood functions.

B. Bellman Backups With Arbitrary LTI State Dynamics

As in previous work in [5], the Bellman backups used so far assume random walk state transitions from s_t to s_{t+1}. On the other hand, many problems such as target search and tracking

[1] The two prior components in this example are evaluated against separate categories for "no detection left" and "no detection right," which together make up a nonconvex "no detection" semantic observation class via an MMS model.

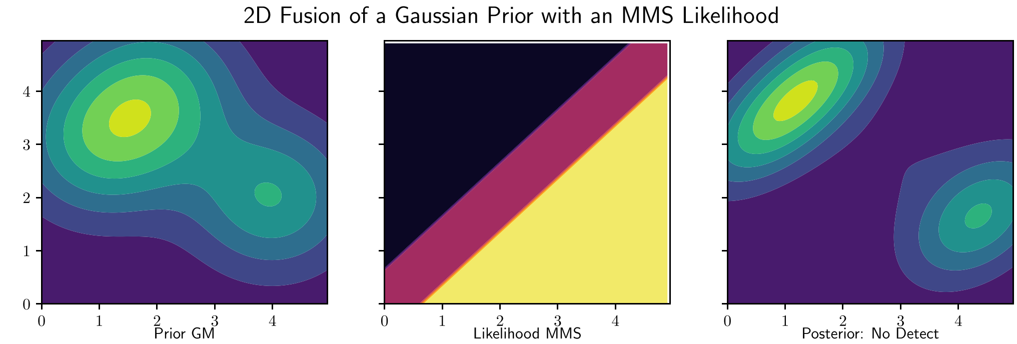


Fig. 3. GM belief update with an MMS observation likelihood model. The negative observation of "No Detection" causes the posterior to split further into a bimodal distribution.

require modeling more sophisticated dynamic behaviors, e.g., via LTI state-space models. In discrete time, such dynamics are represented by a state transition matrix (STM) $F \in \mathbb{R}^{N \times N}$ and action effect $\Delta(a_t) \in \mathbb{R}^N$, such that

$$s_{t+1} = Fs_t + \Delta(a_t).$$

In this case, the state transition PDF takes the form

$$T = \phi(s_{t+1} | Fs_t + \Delta(a_t), \Sigma^a).$$

Using this altered transition model to rederive (30), the new intermediate alphas are

$$\alpha^i_{a,o}(s) \approx \sum_{h=1}^{M} w_h \phi(Fs | \hat{\mu}_h - \Delta(a_t), \hat{\Sigma}_h + \Sigma^a).$$

However, CPOMDP policy approximation requires that the α functions depend on the "current" state s, not on mappings of s. Therefore, the Gaussian dependency on Fs must be converted to a dependency on s

$$\phi(Fs|\mu, \Sigma) \propto \phi(s|\tilde{\mu}, \tilde{\Sigma}).$$

Expanding the left-hand side of this last expression

$$\phi(Fs|\mu, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(Fs - \mu)^T \Sigma^{-1} (Fs - \mu)\right)$$

the STM can be factored within the exponential[2]

$$\phi(Fs|\mu, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(s - F^{-1}\mu)^T F^T \Sigma^{-1} F (s - F^{-1}\mu)\right).$$

The exponential term then resembles a Gaussian

$$\phi(s|\tilde{\mu}, \tilde{\Sigma}), \ \ \tilde{\mu} = F^{-1}\mu, \ \ \tilde{\Sigma} = F^{-1}\Sigma F^{-T}.$$

To address the normalization in front of the exponential, a weighting term ω is introduced

$$\omega = |F^{-1}F^{-T}|^{-\frac{1}{2}}.$$

Multiplying and dividing by ω gives

$$\begin{aligned}\phi(Fs|\mu, \Sigma) &= \frac{\omega}{\omega}|2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(s - F^{-1}\mu)^T F^T \Sigma^{-1} F (s - F^{-1}\mu)\right) \\ &= \frac{1}{\omega}|2\pi F^{-1}\Sigma F^{-T}|^{-\frac{1}{2}} \\ &\quad \times \exp\left(-\frac{1}{2}(s - F^{-1}\mu)^T F^T \Sigma^{-1} F (s - F^{-1}\mu)\right).\end{aligned}$$

Finally, a weighted Gaussian can be recognized as

$$\phi(Fs|\mu, \Sigma) = \frac{1}{\omega}\phi(s|\tilde{\mu}, \tilde{\Sigma})$$

$$\to \alpha_{a,o}(s) \approx \sum_{h=1}^{M} w_h \cdot \frac{1}{\omega} \cdot \phi(s|\tilde{\mu}_{h,a}, \tilde{\Sigma}_a).$$

These equations reduce to the original VB-POMDP backup equations when $F = I$ (identity). This transformation can also be applied to the original CPOMDP approximation with unnormalized GM observation models [5].

IV. Clustering-Based GM Condensation

The number of GM mixands for α functions and $b(s)$ can still become significantly large over iterations/time even with the VB approximation. This section describes a novel GM condensation algorithm to help reduce the computational overhead and enable faster policy computation and online belief updates. Numerical studies comparing the effectiveness of different Gaussian clustering metrics are also presented, showing that a Euclidean distance measure between Gaussian means provides the best overall balance between computational speed and accuracy in terms of speeding up the widely used Runnalls' condensation algorithm [31] for large GMs. The Runnalls' algorithm uses upper bounds on the KLDs between uncondensed GMs and condensed GMs to select successive pairs of mixands for mixture moment-preserving mergers, and as such is better able to retain information from uncondensed GMs compared to other similar condensation methods [32], [50] while also requiring little additional computational overhead.

[2] Assuming F is invertible; this is always the case for LTI systems since F comes from the corresponding matrix exponential.

Algorithm 2: Clustering-Based Condensation Algorithm.

```
Input: Mixture, $K$, $\psi$
Clusters = K-means(Mixture,K);
Create empty NewMixture;
for $C \in$ Clusters do
    $\hat{C}$ = Runnalls($C$, floor($\frac{\text{size}(C)K\psi}{\text{size(Mixture)}}$));
    NewMixture.add($\hat{C}$);
end for
return NewMixture;
Subfunction: Runnalls
Input: $C$, max
while size($C$) > max do
    for unnormalized Gaussians $G_i, G_j \in C$ do
        $[w_{i,j}, \mu_{i,j}, \Sigma_{i,j}] =$ Merge($G_i, G_j$);
        Compute KL divergence upper bound:
        $B_{ij} = \frac{1}{2}[(w_i + w_j)\log|\Sigma_{i,j}|$
         $- w_i \log|\Sigma_i| - w_j \log|\Sigma_j|]$;
        $G_i =$ Merge($G_i, G_j$), where $(i, j) = \arg\min B_{ij}$;
        $C$. remove($G_j$);
    end for
end while
return $C$;
Subfunction: Merge
Input: $G_i, G_j$
$w_m = w_i + w_j$
$\mu_m = \frac{w_i}{w_m}\mu_i + \frac{w_j}{w_m}\mu_j$
$\Sigma_m = \frac{w_i}{w_m}\Sigma_i + \frac{w_j}{w_m}\Sigma_j + \frac{w_i w_j}{w_m}(\mu_i - \mu_j)(\mu_i - \mu_j)^T$
return $w_m, \mu_m, \Sigma_m$
```

A. Clustering-Based Condensation Algorithm

To remain computationally tractable, the GMs representing each α function must also be condensed such that

$$\alpha_n^i = \sum_{k=1}^{M} w_k \phi(s|\mu_k, \Sigma_k) \approx \hat{\alpha}_n^i = \sum_{k=1}^{\tilde{M}} \hat{w}_k \phi(s|\hat{\mu}_k, \hat{\Sigma}_k)$$

where $\tilde{M} < M$ (mixture terms in $b(s)$ must also be compressed following dynamics prediction and Bayesian observation updates). Existing GM condensation algorithms perform myopic pairwise merging of the M components in α_n^i, such that the resulting $\tilde{M}$ components in $\hat{\alpha}_n^i$ minimize some information loss metric [4], [5]. Naïve pairwise merging tends to be very expensive and slow when $M \geq 100$ (which is often the case for long horizon Bellman recursions with $N \geq 2$).

To improve condensation speed, a novel "divide and conquer" strategy is employed, which first preclassifies the mixture indices into K local clusters (submixtures), and then condenses each cluster to some predetermined number of components ψ via pairwise merging, before recombining the results to a condensed mixture with the desired size $\tilde{M} < M$. For merging within submixture clusters, the Runnalls' algorithm [31] is used, which uses an upper bound on the KL divergence between the premerge and postmerge submixture to select the least dissimilar component pairs merging. This process is outlined in Algorithm 2.

Since submixtures may have different precondensation sizes depending on the clustering method used, this approach is prone to overcondensation when each submixture is naively condensed to the same final size. To avoid this, each submixture is condensed according to the proportion of mixands it contains with respect to the original mixture. This means a submixture containing h mixands would be condensed to $\psi = \text{floor}(\frac{h\tilde{M}}{M})$. This can still result in overcondensation if $\frac{h\tilde{M}}{M}$ is not an integer for at least one submixture, but the difference between the desired size and the resulting size is strictly upper-bounded by the chosen number of submixtures.

Empirical tests indicate that this new hybrid method achieves approximately the same accuracy for condensation performance as classical full scale pairwise merging, although the hybrid method is considerably cheaper and faster (e.g., 22.16 versus 5.69 s for $M = 400 \to \tilde{M} = 20$ with $N = 2$, in Python on a 2.6 GHz Intel i7 processor running Windows 10 with 16 GB of RAM).

Fig. 4 shows a comparison of the classical full-mixture Runnalls' condensation method to our hybrid cluster-then-condense method for a GM with $M = 400$ components, with $K = 4$ and $\psi = 5$. The integral square difference (ISD) metric [50] is used to assess the accuracy of each method, where, given two GMs $\text{GM}_h(s)$ and $\text{GM}_r(s)$

$$\begin{aligned}
&\text{ISD}[\text{GM}_h(s), \text{GM}_r(s)] \\
&\quad = \int_{\mathcal{S}(S)} (\text{GM}_h(s) - \text{GM}_r(s))^2 ds = J_{hh} - 2J_{hr} + J_{rr}
\end{aligned}$$

$$J_{hh} = \sum_{i=1}^{N_h}\sum_{j=1}^{N_h} w_i w_j \phi(\mu_i|\mu_j, \Sigma_i + \Sigma_j)$$

$$J_{hr} = \sum_{i=1}^{N_h}\sum_{j=1}^{N_r} w_i w_j \phi(\mu_i|\mu_j, \Sigma_i + \Sigma_j)$$

$$J_{rr} = \sum_{i=1}^{N_r}\sum_{j=1}^{N_r} w_i w_j \phi(\mu_i|\mu_j, \Sigma_i + \Sigma_j).$$

This example indicates that both methods result in condensed GMs that have approximately the same ISD compared with the original GM, although the hybrid cluster-then-condense method is considerably faster.

B. Empirical Clustering Metric Comparisons

Theoretically, the cluster-then-merge approach is natural to consider, since any GM can be generally viewed a "mixture of local submixtures." From this standpoint, mixture components belonging to different local submixtures are unlikely to be directly merged in a pairwise global condensation algorithm, whereas those belonging to the same submixture are more likely to be merged. The global merging operation can then be broken up into several smaller parallel merging operations within each submixture. In our initial approach, the submixtures are identified using a simple fast k-means clustering heuristic on the component means. Additional work verifies the robustness of

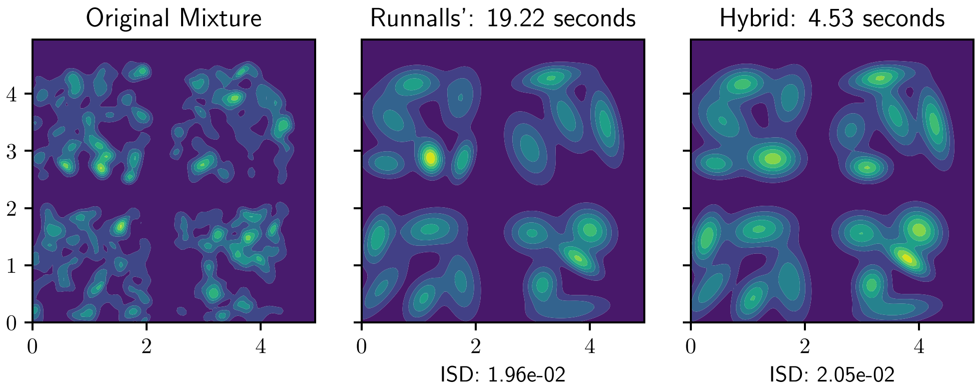


Fig. 4. Condensation comparison of Runnalls' method to preclustering hybrid method: an initial mixture of 400 mixands is condensed to 20 mixands; the hybrid method results in a similar ISD as Runnalls' alone, but significantly faster.

this method for general problem settings, and other techniques for identifying submixture groups could also be used (e.g., to also account for mixand covariances, etc.).

The k-means clustering heuristic employed in the example above utilized the Euclidean distance between mixand means. While this metric results in simple fast clustering, it also underutilizes the information available. Alternative techniques for clustering were therefore also evaluated; these take into account additional information, specifically mixand covariances, with the goal of finding a method that performs with an improved level of accuracy without sacrificing too much of the speed achieved by the Euclidean distance between means. Five methods in total were considered for submixture formation: four alternative PDF distance measures and the original Euclidean distance heuristic. Each alternative method chosen has a closed-form derivation for normalized Gaussian PDFs, and utilizes only the mixand mean and covariance. Weights are considered within the second part of the procedure when Runnalls' method is used to combine similar mixands.

The first alternative distance is the symmetric KLD, which measures the difference in expectation between two distributions. The symmetric KLD is defined for two normal distributions G_i and G_j as

$$D_{\text{sym}KL} = \frac{\text{KLD}(G_i||G_j) + \text{KLD}(G_j||G_i)}{2}.$$

Next, the Jensen–Shannon divergence is considered. The Jensen–Shannon divergence is a symmetric and smoothed version of KLD that uses an average of the two distributions G_i and G_j

$$\text{JSD}(G_i||G_j) = \frac{1}{2}\text{KLD}(G_i||M) + \frac{1}{2}\text{KLD}(G_j||M)$$

$$\text{where } M = \frac{1}{2}(G_i + G_j).$$

The two-Wasserstein distance, sometimes referred to as the earth mover's distance (EMD), is a measure of the minimum cost of turning one distribution into the other, factoring in both distance between distributions and the probability mass of each. The two-Wasserstein distance is defined as

$$W_2(G_i, G_j)^2 = ||\mu_i - \mu_j||_2^2 + Tr(\Sigma_i + \Sigma_j - 2(\Sigma_j^{1/2}\Sigma_i\Sigma_j^{1/2})^{1/2}).$$

Finally the Bhattacharyya distance is considered, which measures overlap between two distributions and is also closely related to the Hellinger divergence. This takes into account both distance between means and similarity of covariances. The Bhattacharyya distance is defined as

$$D_B = \frac{1}{8}(\mu_1 - \mu_2)^T\Sigma^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\left(\frac{|\Sigma|}{\sqrt{|\Sigma_1||\Sigma_2|}}\right)$$

$$\text{where } \Sigma = \frac{\Sigma_1 + \Sigma_2}{2}.$$

To more directly compare tests of different dimensions, starting sizes, and ending sizes, here we use the normalized version of the ISD metric. The normalized ISD [53] constrains each measurement to a range NISD $\in [0, 1]$, and is derived from the ISD definition as

$$\text{NISD}[\text{GM}_h, \text{GM}_r] = \sqrt{\frac{\text{ISD}[\text{GM}_h, \text{GM}_r]}{(J_{hh} + J_{rr})}}. \tag{33}$$

Test mixtures in $N = 1, 2,$ and 4 dimensions were generated by sampling means from a uniform distribution from 0 to 10 on $\mathbb{R}^N$, sampling covariances from a Wishart distribution with N degrees of freedom and a matrix prior of identity scaled by a factor of 2, and sampling weights from a uniform distribution from 0 to 1. Each combination of dimensionality, clustering method, number of starting mixands, number of clusters, and final mixture size was repeated on ten different randomly generated mixtures. The time for clustering and condensation, and the accuracy of clustering and condensation, characterized by the normalized ISD between the starting and final mixtures, were recorded. Additionally, Runnalls' method without clustering was used as an state-of-the-art baseline for accuracy, and time and normalized ISD for Runnalls' were recorded. The time and normalized ISD results for each distance measure were then able to be compared to one another and to the time and normalized

TABLE I
TIME AND ACCURACY VERSUS RUNNALLS' METHOD

	Sym KLD	JSD	Euclid	EMD	Bhatt
	Norm. ISD ratio				
Dimension					
1	0.4527	0.6656	1.0666	1.0675	0.6067
2	1.7338	2.0567	1.9774	2.0408	1.9666
4	1.4089	2.1097	1.7130	5.0584	2.4256
	Time ratio				
1	2.3470	1.5198	0.1783	0.7512	0.9989
2	5.0418	2.0694	0.1725	0.6476	1.5415
4	14.6303	6.8890	0.1835	2.8365	4.2750

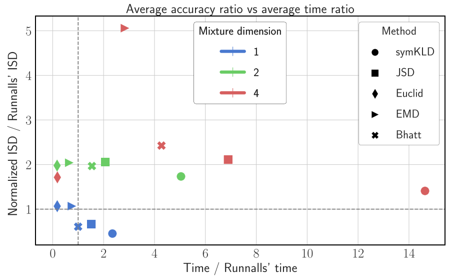


Fig. 5. Time and normalized ISD of clustering-based condensation compared to Runnalls' method without clustering.

ISD results achieved using Runnalls' method. The results were obtained in Python on a 3.3 GHz Intel i7 processor running Ubuntu 16.04, with 32 GB of RAM.

Table I presents the results for time of clustering and condensation and the accuracy of each method as a percentage of the Runnalls' time and accuracy, averaged across all parameters barring dimension. These results are also presented graphically in Fig. 5. In general, the alternative methods compared poorly to Euclidean distance in the accuracy versus speed tradeoff. In higher dimensions, the alternative methods tended to heavily favor circular or hyper-spherical clusters, leading to suboptimal clustering in mixtures with elongated high density regions. This combined with the additional overhead needed to compute the alternative metrics led to the Euclidean distance measure consistently providing the best balance of accuracy versus speed, particularly in higher dimensions. Therefore, the Euclidean distance is used in the remainder of this article.

V. SIMULATION EXPERIMENTS

This section examines application of VB-POMDP to three simulated versions of the target search and localization problem (with $N = 2, 4$), as well as a five-robot simultaneous localization and navigation problem ($N = 10$). VB-POMDP's performance on the target search application is compared to the performance of other state-of-the-art policy approximation methods. The clustering-based GM condensation method is used for all scenarios and with all policy approximations.

A. Colinear Cop/Robber Results

Table II compares the resulting average final rewards achieved over a 100 step simulation for 100 simulation runs, using policy approximations for the 1-D cop–robot search problem presented earlier in Section II-G. The second column shows the average final rewards the proposed VB-POMDP method [with the softmax likelihood model shown in Fig. 1(b)], while the first column shows the average final rewards obtained for the GM-POMDP policy approximation of [5] [using the GM observation models shown in Figs. 1(c)]. Both methods used the hybrid GM clustering technique introduced in Section III-B. Results for a third greedy one-step implementation of the latter approximation are also shown in the third column. While a greedy policy approximation is generally expected to be suboptimal, it provides a realistic minimum implementation cost baseline result for use on a robotic platform, and also provides an indication of the problem's difficulty (i.e., in this case, in a single dimension).

TABLE II
REWARDS ACHIEVED ON BASIC COLINEAR TARGET SEARCH PROBLEM
(STANDARD DEVIATIONS OVER 100 SIMULATION RUNS SHOWN)

Method	Mean Reward	Standard Deviation
VB-POMDP	57	±54
GM-POMDP	59	±30
Greedy	19	±57

All methods were compared pair-wise using the Student's *t*-test for the difference of two means, with 100 samples each. Statistically, the VB-POMDP policy approximation average performance could not be differentiated from the baseline GM-POMDP policy, with $p > 0.05$. However, both policies achieved a significantly higher average accumulated reward than the comparison greedy approach, with $p < 0.05$. These results indicate that the VB-POMDP approximation performs as well as the GM-POMDP approximation on this problem. The VB approximations described earlier therefore do not lead to any significant compromises in optimality for this problem compared to the state of the art.

B. Two-Dimensional (2-D) Random Walk Robber Results

Extending the colinear search problem, the cop robot now attempts to localize and intercept the robber in a bounded 2-D space $S = \mathbb{R} \times \mathbb{R}$, where $s_{c,t} = [\text{Cop}_{x,t}, \text{Cop}_{y,t}]^T$ and $s_{r,t} = [\text{Rob}_{x,t}, \text{Rob}_{y,t}]^T$. The robber again executes a Gaussian random walk

$$s_{r,t+1} \sim \mathcal{N}(s_{r,t}, I).$$

The cop's noisy actions are $A = \{\text{East, West, North, South, Stay}\}$; each has an expected displacement of 1 m in the corresponding direction. The cop receives semantic observations $\Omega = \{\text{East, West, North, South, Near}\}$, simulating a coarse proximity sensor that depends on the relative location between the cop and robber; the softmax likelihood model for this is shown in Fig. 6. Rewards are based on the cop's distance from the robber

$$r(\text{dist}(\text{Rob}_t, \text{Cop}_t) \leq 1) = 5$$
$$r(\text{dist}(\text{Rob}_t, \text{Cop}_t) > 1) = 0.$$

As such, policies are found for the combined difference state $s_t = s_{r,t} - s_{c,t} = [\Delta X_t, \Delta Y_t]^T = [\text{Rob}_{x,t} - \text{Cop}_{x,t}, \text{Rob}_{y,t} -$

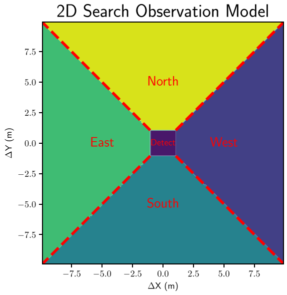


Fig. 6. Softmax semantic observation model for 2-D search problem, with $s = [\Delta X, \Delta Y] = [\text{Rob}_x - \text{Cop}_x, \text{Rob}_y - \text{Cop}_y]$.

TABLE III
AVERAGE FINAL REWARDS FOR THE 2-D SEARCH PROBLEM (STANDARD DEVIATIONS OVER 1000 SIMULATION RUNS)

Method	Mean Reward	Standard Deviation
Perfect Knowledge	118.4	±26.5
VB-POMDP	110.8	±25.8
GM-POMDP	101.2	±23.4
Greedy	81.2	±23.7

$\text{Cop}_{y,t}]^T$, so that $N = 2$. The corresponding continuous state reward function is modeled as a GM consisting of a single weighted Gaussian located at the point $[0 - \Delta(a)_x, 0 - \Delta(a)_y]$, where $\Delta(a)$ is the expected displacement of the cop resulting from a given action. This incentivizes the cop to drive s_t to [0,0]. Of note, no negative reward is introduced as a time penalty. With a single source of positive reward and no negative rewards, the actual weight of the reward GM mixands is irrelevant, as any reward gradient is enough to encourage the cop to maximize reward by reaching the desired state quickly.

Both GM-POMDP and VB-POMDP solvers were given 8 h to find policies, though in both cases approximations had converged within 4 h. In addition to comparing these approximations to a simple greedy policy as a reference baseline "online" approximation as before, an omniscient "Perfect Knowledge" solver (i.e., which has access to perfect observations about the robber's location) was also assessed to provide an upper bound on optimal policy performance. This solver leads to a policy whereby the Cop's actions minimize its distance from the mode of its belief in the robber's location, where the belief is modeled by a Dirac delta function centered on the robber's true state.

All policy approximation methods were again compared pairwise using the Student's t-test for the difference of two means, with 1000 samples each. Table III shows the mean reward and standard deviation for the results, and Fig. 7 show the accumulated results, in which the VB-POMDP method significantly outperforms both the GM-POMDP and greedy approximations with $p < 0.05$, while it is outperformed by the perfect knowledge policy with $p < 0.05$.

To examine the policies' sensitivities to the problem parameters, the simulations were repeated with slower robber dynamics

$$s_{r,t+1} \sim \mathcal{N}(s_{r,t}, 0.7 \cdot I).$$

The results of these tests are shown in Fig. 8. The relationship between perfect knowledge, VB-POMDP, and GM-POMDP remains unchanged. The Greedy policy now outperforms GM-POMDP, yet still falls short of VB-POMDP, both with $p < 0.05$. These results, when combined with those of Fig. 7 indicate that problems with higher levels of noise or uncertainty derive greater benefit from more complex planning and control algorithms.

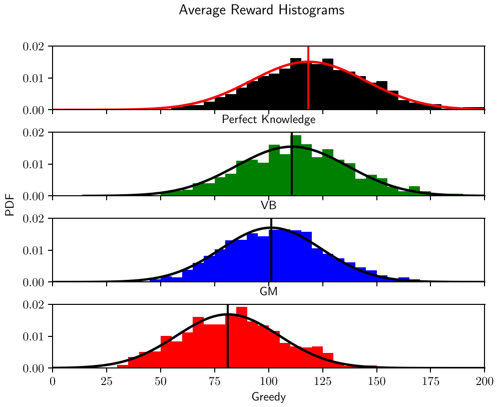


Fig. 7. Total reward histograms for 2-D search problem.

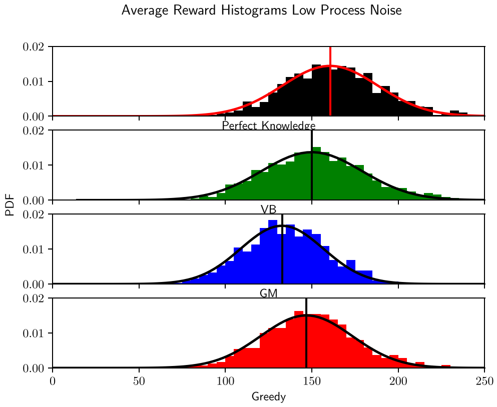


Fig. 8. Total reward histograms with slower robber.

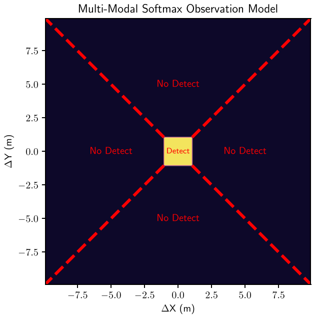


Fig. 9. MMS semantic observation model for 2-D search problem, with $s = [\Delta X, \Delta Y] = [\text{Rob}_x - \text{Cop}_x, \text{Rob}_y - \text{Cop}_y]$.

To examine the algorithms' responses to differing observation models, the tests were run again with a modified version of the 2-D search observation model, shown in Fig. 9. This model contains two observations, the "Detect" observation, which is

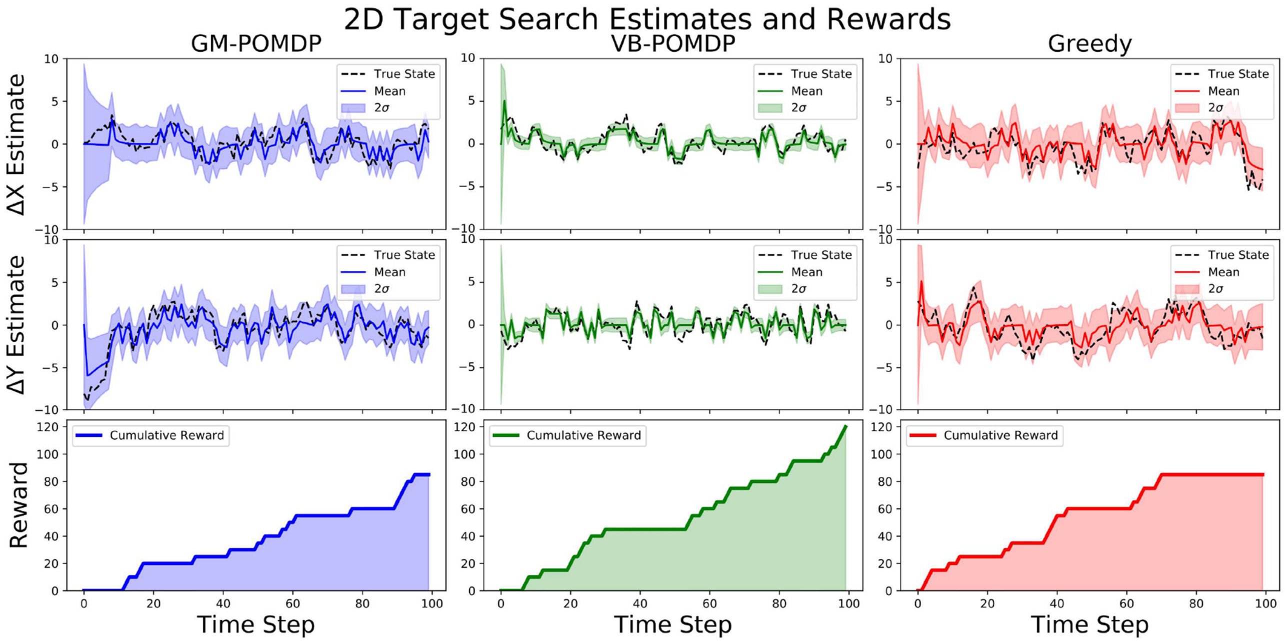


Fig. 10. Comparison of robber state estimates and rewards for a typical 2-D Target Search simulation using different policy approximations. VB-POMDP (green) maintains a slightly overconfident belief but avoids extended periods without reward, leading to a higher average reward than either GM-POMDP method (blue) or Greedy (red). All distances in meters, with each $\Delta T = 1$ s time step representing a single discrete simulated dynamics and measurement update.

TABLE IV
CAPTURE STATISTICS FOR MMS 2-D SEARCH

Method	Percent Caught	Mean Steps to Catch	SD
VB-POMDP	65.3	48.4	±26.9
GM-POMDP	51.2	43.2	±27.3
Greedy	35.3	49.3	±29.4

identical to that from Fig. 6, and the "No Detect" observation, which combines the "North, South, East, and West" observations of Fig. 6. This leads to an MMS observation modal and highly non-Gaussian state beliefs with a generally lower certainty of the target's position. The measured statistic of these tests is the number of steps for the pursuer to catch the target for the first time. Each simulation was allowed 100 steps to reach this goal state before termination.

The results of these tests are shown in Table IV. A binomial statistical test was used to compare the percentage of captures for each method. From this metric, it is once again found that VB-POMDP outperforms both GM-POMDP and Greedy policies in pair-wise statistical comparisons with $p < 0.05$, while GM-POMDP outperforms Greedy with $p < 0.05$.

C. Discussion

The colinear search scenario results indicate that for a simple problem VB-POMDP achieves near parity with GM-POMDP method, and that both methods surpass the greedy approach. This is expected, as both the GM and softmax observation models were constructed to approximate the same semantic model with all else held equal. Importantly, the VB approximations used by VB-POMDP do not seem to significantly impact the quality of the policy approximation.

Comparing the results from Section V suggests that the VB-POMDP approximation outperforms GM-POMDP as problem complexity increases. A contributing factor to this disparity is that VB-POMDP can complete more backup steps within an allotted time than GM-POMDP for the 2-D search problem (e.g., VB-POMDP completed six times more backups than GM-POMDP for this problem parameterization running on a 2.6 GHz processor running Linux with 16 GB RAM). This is largely due to the number of mixands generated by each method. In a single backup step, the GM-POMDP method produces alpha-functions of size $|\alpha_n| = \sum^{|\Omega|} |\alpha_{n-1}| M_o$ [where M_o is the number of unnormalized GM terms needed to define $p(o'|s')$ in (3)], whereas VB-POMDP produces alpha-functions of size $|\alpha_n| = \sum^{|\Omega|} |\alpha_{n-1}|$. The additional time needed for VB to converge is more than offset by the condensation time savings from having fewer mixands. An example of the results of these time savings is shown in Fig. 10. The VB-POMDP policy allows the Cop to act more strategically than the GM-POMDP policy when it loses contact, while avoiding the naive pursuit strategy of the Greedy method. This agrees with intuition: completing additional backups should allow the solver to better approximate the optimal policy.

Another contributing factor to the quality of the VB-POMDP approximation is the amount of condensation required between backups. As shown in Fig. 5, condensation of larger mixtures leads to larger approximation errors, setting a practical limit on how closely a theoretically optimal policy can be approximated with a finite number of GM components. Since VB-POMDP generates a smaller number of mixands during the backup step, such errors from condensation accumulate less often. This suggests VB-POMDP and GM-POMDP ought to produce the same results if each were given an infinite amount of time and an unbounded number of mixands for policy approximation. However, in practical resource limited situations for offline computation, VB-POMDP holds a distinct advantage.

Note that the VB approximation for state belief updates in (31) can produce slightly overconfident posteriors, as seen in Fig. 10. As discussed in [3], this can be mitigated by using a VB importance sampling approximate softmax update, which carries an additional Monte Carlo importance sampling step to compensate for optimistic covariances produced by the VB softmax update approximation. This would result in an additional speed versus accuracy tradeoff.

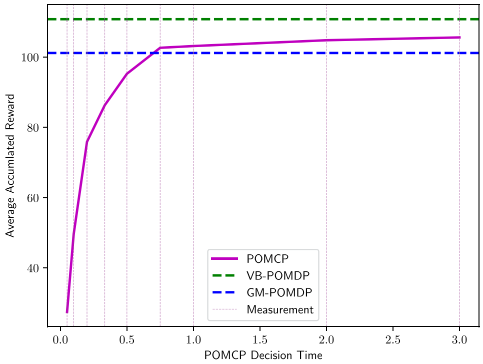


Fig. 11. Average POMCP final rewards versus planning time.

D. Comparison to Online Algorithms

Like GM-POMDP, VB-POMDP is an offline policy approximation algorithm, requiring the majority of computation to take place prior to deployment on a real platform. Alternative online policy approximation methods could also be used to solve CPOMDPs, but would result in different implementation tradeoffs for speed, online computing requirements, and belief space coverage. This section compares VB-POMDP to a state-of-the-art online approach known as POMCP [25], which is a Monte Carlo Tree Search [40] based algorithm for online POMDP approximation. While this comparison of offline VB-POMDP to online POMCP is not strictly an "apples to apples" comparison, it does provide some useful insight to underscore the practicality of VB-POMDP (and offline solvers more generally) for problems like dynamic search and localization with semantic observations.

POMCP uses a generative model of state dynamics and observations to propagate a search tree of histories, choosing the path through the tree with the greatest expected reward. POMCP is of particular interest since it has successfully been applied to problems with large discrete state and observation spaces beyond what many nonsampling based offline algorithms can typically handle. Due to the fact that it only requires a "black-box" generative model of the problem to function, POMCP has also been shown to function well in continuous state spaces [42]. POMCP also provides an online "anytime algorithm," where computation can be cut short at some threshold and return the best answer found to that point. POMCP is considered here as a baseline state-of-the-art "general purpose" policy solver, though it can suffer from extremely suboptimal worst case behavior in certain kinds of problems with sparse rewards (as in search/localization), due to its reliance on the upper confidence tree (UCT) algorithm [48].

The simulations here make use of the Julia POMCP implementation in the POMDPs.jl toolbox [54]. Simulations of the 2-D target search problem were run for nine separate test cases using a 3.3 GHz processor, 32 GB of RAM, and a Julia language implementation on a system running Ubuntu 16.04. The cases were run with an exploration parameter of $c = 10$, and were allowed a planning depth of up to 100 time steps, mirroring the 100 steps allowed during the each run. The solver was allowed as many tree queries as could be completed within a given decision time. The cases differed only in the amount of time was allowed for an action to be chosen, varying from 0.05 s up to 3 s. These bounds were chosen to compare to the typical online decision time required for VB-POMDP or GM-POMDP running in a Python environment on the same machine, 0.05 s, up to a maximum allowable wait time for a physical robot preforming a real-time task, 3 s. Each case was run for 100 trials, with the mean final rewards shown in Fig. 11 (results were similar for more than 100 trials).

From Fig. 11, it is clear that increasing allowable decision time beyond 1 s yields real but diminishing returns. Compared to offline GM-POMDP, POMCP reaches statistical similarity in about 0.5 s of decision time ($p > 0.05$). Compared to VB-POMDP, POMCP reaches statistical similarity around 3 s of decision time ($p > 0.05$). While it is likely that POMCP would continue to show marginal improvements with additional decision time, it should be noted that these simulations were run with vastly more computing power than would be available to a typical small mobile robotic platform, and were run in isolation without siphoning off available processing for tasks such as control, vision, or communication. While Fig. 11 demonstrates that an online solver such as POMCP can achieve similar results to a full-width offline solver if given sufficient resources, the results imply that offline approximations like VB-POMDP can offer some implementation advantages for mobile robotic platforms.

E. LTI Dynamics Models Simulations

The 2-D search problems considered thus far used kinematic nearly constant position (NCP) random walk transition models for the robber, with $F = I$. The problem is extended so that the robber now uses a kinematic NCV model, which is commonly used for target search and tracking. The NCV model requires four states to capture differences in cop/robber position and velocities. For a given action and an augmented state vector $s = [\Delta X, \Delta Y, V_{\Delta X}, V_{\Delta Y}]$, where $V_{\Delta X}$ and $V_{\Delta Y}$ are relative

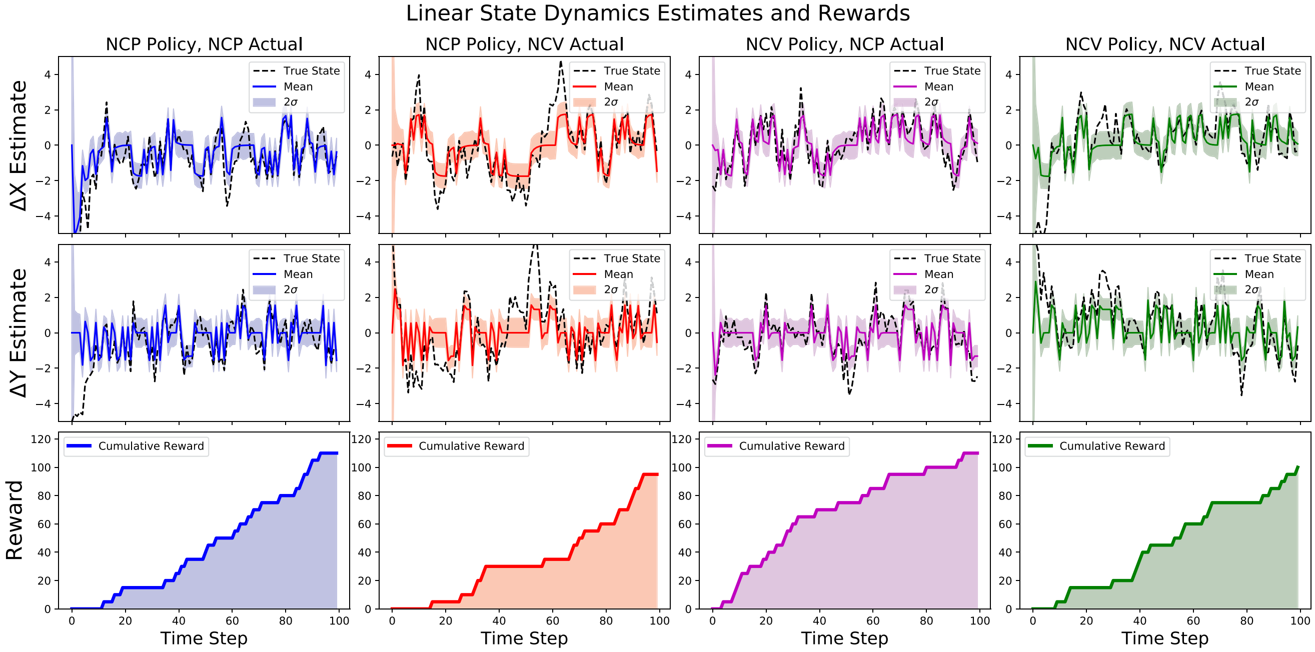


Fig. 12. Comparison of state estimates and rewards for a typical run of the dynamic 2-D target search problem under NCP and NCV models. All runs use VB-POMDP, which adapts in mismatched robber model cases to achieve nearly the same performance obtained by policies using the correct robber model. All distances in meters, with each time step $\Delta T = 1$ sec representing a single discrete simulated dynamics and measurement update.

TABLE V
AVERAGE FINAL REWARDS FOR NCP AND NCV POLICIES WITH DIFFERENT ACTUAL TARGET MODELS

	NCP Actual	NCV Actual
NCP Policy	110.0 ±29.3	97.5 ±26.2
NCV Policy	110.7 ±26.8	99.3 ±24.2

distance rates of change

$$s_{t+1} = Fs_t + \Delta(a_t) = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} s_t + \Delta(a_t)$$

where ΔT is the physical time step. This state dynamics model requires the generalized Bellman backup equations for LTI dynamics introduced in Section III-B. The softmax models used previously for semantic position observations are easily updated to accommodate the velocity states, namely by augmenting all softmax class weights and biases with additional rows of 0's for the new state dimensions.

The target search simulations considered so far constrained the policy approximation's robber state dynamics model to be identical to the robber's actual dynamics. However, the true dynamics model will not always be exactly known in real applications. This constraint was therefore relaxed to examine VB-POMDP's sensitivity to model mismatches in this higher dimensional setting. Policies were approximated with VB-POMDP assuming either an NCP or NCV model; each policy type was then implemented in scenarios where the robber either actually used the NCP model or NCV model. Table V shows the results of 100 simulations for each scenario.

The VB-POMDP policy is able to adapt to which ever transition model is actually being used by the robber. While scenarios in which the policy and actual robber model matched performed slightly better on average, the mismatched model scenarios still achieved similar results. As seen in the example in Fig. 12, the consequences of model mismatch are more apparent when the simpler NCP model is assumed for more complex NCV actual robber dynamics. In the "NCP Policy, NCV Actual" scenario, the approximate policy assumes zero mean robber velocity at all times. This repeatedly leads to incorrect beliefs, causing the policy to select suboptimal actions, which lead to a slightly lower final reward. This issue is less pronounced in the "NCV Policy, NCP Actual" simulations, where the belief converges to a correct estimate of zero mean velocity, and achieves similar rewards to the policy trained on NCP models. These results show that the approximations used by VB-POMDP perform well in a higher dimensional target search setting and still lead to reasonable behaviors even with slight model mismatches.

F. *Multirobot Localization/Goal-Seeking Problem*

The "Cop and Robber" problems considered so far in $N = 2, 4$ continuous state dimensions still use a fairly limited set of actions and observations for a single decision-making agent. This section describes a considerably more challenging localization application to assess VB-POMDP's usefulness for approximating policies for CPOMDPs featuring higher dimensional continuous dynamical state spaces and more complex action/observation spaces. The problem consists of five independently

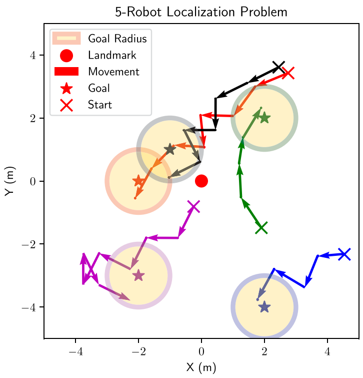


Fig. 13. Set up and policy execution for 5-robot localization.

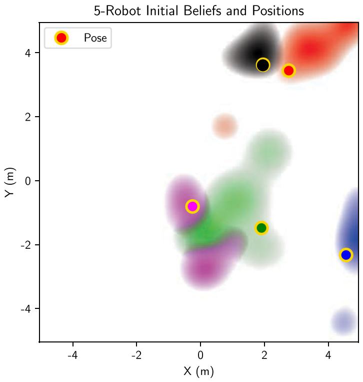


Fig. 14. Initial marginal state PDFs (color-coded by robot).

controlled robotic agents who attempt to reach designated goals in a 2-D plane. The combined localization and movement problem thus contains ten continuous state dimensions. At each time step, only one robot is allowed an action. Each robot can take one of four actions to move (one at a time) in one of the four cardinal directions, resulting in 20 total possible actions. At each time step, a single robot is chosen to move and receive a semantic observation. One of the agents (Robot 1) can semantically observe its location with respect to a fixed landmark. Robots 2–5 can only observe their location with respect to the previously numbered agent (e.g., Robot 3 observes that it is North of Robot 2; Robot 4 observes it is East of Robot 3, etc.). After moving, each robot also receives a semantic observation indicating whether or not it has arrived at its goal (thus providing a source of negative information). All observations and beliefs are processed by a single centralized planner policy, which determines which robot to move at any given time. Fig. 13 shows the problem setup. Fig. 14 depicts an example initial 10-dimensional state belief for this problem, which is highly non-Gaussian and modeled with a GM PDF.

In addition to featuring non-Gaussian initial state beliefs, this problem is made more challenging by the random "dropping" of observations (e.g., simulating the effect of a noisy communication channel between the robots and the centralized planner) and multimodal/non-Gaussian state transition models. At any given time step, there is a uniform probability that one of the observations recorded by the robots will not be received by the planner. Movement actions are subject to multimodal process noise, modeled by the GM process

$$p(s_{t+1}|s_t, a_t) = \frac{1}{3}\sum_{h=1}^{3} p_h(s_{t+1}|s_t, a_t) \tag{34}$$

where mixand $h = 1$ has a zero mean process term and mixands $h = 2, 3$ have nonzero mean process terms to model disturbances perpendicular to the intended direction of travel (e.g., due to wind, slippage, etc.). Using the softmax synthesis methods detailed in [27] and [52], semantic observation models for each robot were separately constructed in two dimensions for Robot 1, and four dimensions for Robots 2–5, before being uniformly extended through the remaining 8/6 dimensions. This extension requires padding the weight vectors for each softmax class with 0's for each added dimension. Robot 1's observation model is identical to the one used earlier in the 2-D search problem, but with absolute coordinates instead of relative ones. For Robots 2–5, the nonzero softmax weight terms are selected to embed a four-dimensional probabilistic parallelepiped, such that for any given location of Robot i, the observation model for Robot $i + 1$ resembles Robot 1's absolute measurement model after a coordinate shift to a given location. These softmax models are also easily modified to derive the MMS models for each robot's goal-relative semantic observations (akin to Fig. 9).

The VB-POMDP policy approximation for this problem efficiently maneuvers each robot to its goal.[3] In contrast to a simple greedy policy, which attempts to move each robot directly to its goal, the VB-POMDP policy pursues information gathering actions by moving robots across multiple semantic observation class boundaries in order to firmly localize positions (see motion traces in Fig. 13). This behavior is further illustrated in Fig. 15. The VB-POMDP policy also strategically positions well-localized robots so that downstream observers are better localized via relative observations. As expected, the VB-POMDP policy begins taking greedy "go directly to goal" actions only after all robots are well-localized. VB-POMDP's ability to produce such sophisticated information gathering behaviors in high-dimensional problems with complex uncertainties underscores its value as a scalable policy approximation for CPOMDPs with semantic observation models.

Note that while straightforward geometric reasoning can be used to easily design softmax observation models for this problem, synthesis of similar likelihood models via unnormalized GMs is especially cumbersome and impractical. At a minimum, thousands of mixands would be needed to define a GM likelihood that accurately models the boundaries of each semantic observation region in the 10-dimensional state space with sufficient resolution, while also obeying the "sum to 1" constraint over all semantic class labels. Such enormous GM likelihoods would make offline Bellman backups for GM-POMDP extremely expensive to implement and Bayesian state PDF belief updates impractical for online execution (even on powerful modern

[3] New policies must be solved for new sets of goals; the policy for results shown here required 36 h to compute in the same computing environment used for the lower dimensional search problems.

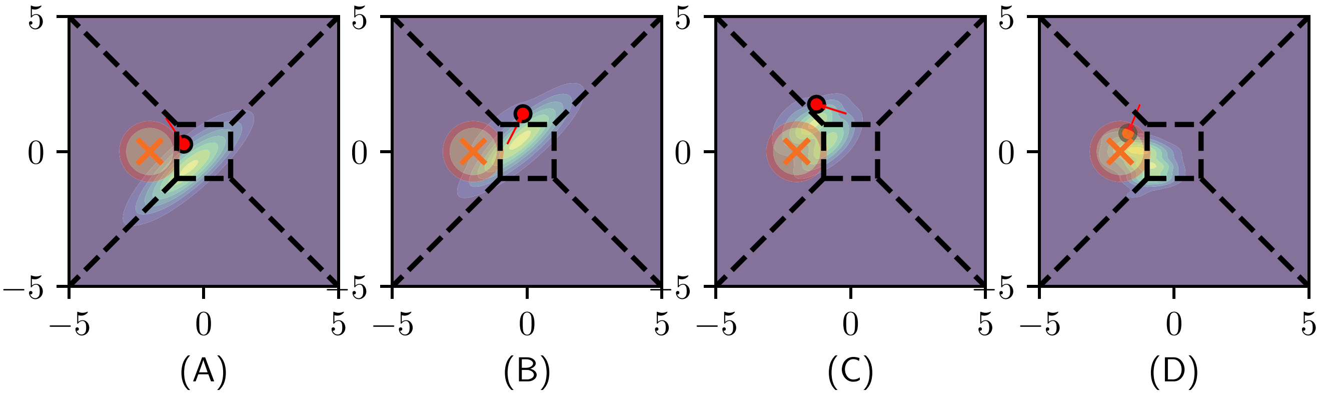


Fig. 15. VB-POMDP policy approximation for the 5-Robot problem displays non-myopic behavior to enable better robot localization before moving robots towards goals: (A), robot (red dot) has a broad uncertainty about its state, shown in the belief heatmap contour; (B) and (C): instead of attempting to move directly to the goal ("x," with "reached" radius shown as circle), the robot moves (red solid trail) across the probabilistic class boundaries of its goal-relative semantic observation model (black dashes); (D): having localized itself sufficiently, the robot moves to its goal state.

computers), as extreme levels of condensation would be needed to maintain manageable GM PDF beliefs.

VI. Conclusion

This article presented VB-POMDP, a VB policy approximation for solving CPOMDPs with hybrid continuous-to-discrete semantic sensor observation likelihoods that are modeled by softmax models. Softmax models are ideal for modeling semantic observation likelihoods, and are cheaper and simpler to construct and evaluate compared to unnormalized GM functions that have been applied for the same purpose in other CPOMDP policy approximations. To overcome the analytical intractability of using softmax models in standard GM PBVI policy approximations for CPOMDPs, a variational Gaussian inference approximation was developed to maintain the closed-form recursive nature of the GM PBVI approximation. This approach also tends to produce far fewer mixture terms in the intermediate PBVI recursion steps, and thus requires less overall computation to approximate the optimal policy. VB-POMDP was also explicitly extended to problems with LTI state dynamics, allowing a broad set of problems to be addressed by the GM PBVI framework. A novel approach to GM condensation was also described and studied, whereby mixture terms are preclustered into submixtures that are then condensed in parallel. This approach was shown to be considerably faster than conventional global mixture condensation techniques, while achieving similar accuracy.

Experimental simulations for a simple mobile target search and localization problem showed that VB-POMDP performed as well as an alternative GM-based CPOMDP policy approximation method, thus indicating that the approximations used by VB-POMDP do not lead to any significant compromises in optimality versus other state-of-the-art approximations. However, VB-POMDP was shown to be significantly more effective on more complex and higher dimensional variants of the target search and localization problem. Simulations for policies trained on target state transition models differing from the true model showed that VB-POMDP is suitably responsive and robust to instances of model mismatch. Finally, VB-POMDP was shown to scale and perform well on a complex 10-dimensional continuous state multirobot localization/goal-seeking problem, featuring highly non-Gaussian uncertainties as well as a large action and semantic observation space.

The results presented here have many interesting implications for developing and applying autonomous probabilistic planning and control algorithms in hybrid continuous-discrete domains. VB-POMDP retains many desirable properties of other GM-based PBVI policy approximation approaches. Among these is the ability to produce deterministic policy approximations for a given set of tent-pole beliefs, as well as the ability to naturally leverage Gaussian sum filters for Bayesian belief updates in domains featuring complex continuous state dynamics and uncertainties (which have also been shown to be more robust than particle filters for several robotics applications [3], [55]). In addition to the search, localization and goal-seeking applications described here, the CPOMDP framework developed here is being leveraged for cooperative human–robot target search and tracking applications. Building on previous work in [3] and [27] and ongoing work in [7], this will enable semantic "human sensor data" from natural language inputs that can be combined with optimal robotic sensing and motion planning in hardware for tightly integrated human–robot teaming. However, VB-POMDP could also be applied to other problems where discrete semantic observations are naturally available as a function of continuous dynamic state variables, e.g., active semantic mapping/SLAM, tactile reasoning for manipulation, collision avoidance, planning/control of stochastic hybrid dynamical systems, etc.

Various relaxations of modeling assumptions made in this article are also possible. For instance, as discussed in [7], VB-POMDP can be extended to nonlinear state-dependent switching mode dynamics models to accommodate more complex probabilistic state transition PDFs that are modeled with softmax functions [4]. This article also motivates study of complex CPOMDPs with even larger spaces of actions and observations than those considered here, including continuous action spaces. The various algorithmic approximations presented here also warrant further analysis when used in approximating optimal planning. In particular, it is desirable to obtain bounds on the accuracy of the K-means hybrid GM condensation method, as well as possible lower bounds on the value function via the VB inference approximation to establish approximation error bounds with respect to the exact optimal CPOMDP policy.

References

[1] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Coordinated decentralized search for a lost target in a Bayesian world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 1, pp. 48–53.

[2] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 574–584, 2010.

[3] N. R. Ahmed, E. M. Sample, and M. Campbell, "Bayesian multicategorical soft data fusion for human–robot collaboration," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 189–206, Feb. 2013.

[4] E. Brunskill, L. P. Kaelbling, T. Lozano-Pérez, and N. Roy, "Planning in partially-observable switching-mode continuous domains," *Ann. Math. Artif. Intell.*, vol. 58, nos. 3/4, pp. 185–216, 2010.

[5] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous POMDPs," *J. Mach. Learn. Res.*, vol. 7, pp. 2329–2367, 2006.

[6] L. Burks and N. Ahmed, "Optimal continuous state POMDP planning with semantic observations," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 1509–1516.

[7] L. Burks *et al.*, "Closed-loop Bayesian semantic data fusion for collaborative human-autonomy target search," in *Proc. 21st Int. Conf. Inf. Fusion*, 2018, pp. 2262–2269.

[8] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte, "Parametric POMDPs for planning in continuous state spaces," *Robot. Auton. Syst.*, vol. 54, no. 11, pp. 887–897, 2006.

[9] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte Carlo value iteration for continuous-state POMDPs," in *Algorithmic Foundations of Robotics IX*. New York, NY, USA: Springer, 2010, pp. 175–191.

[10] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 895–913, 2011.

[11] E. Zhou, M. C. Fu, and S. I. Marcus, "Solving continuous-state POMDPs via density projection," *IEEE Trans. Autom. Control*, vol. 55, no. 5, pp. 1101–1116, May 2010.

[12] H. Bai, D. Hsu, M. J. Kochenderfer, and W. S. Lee, "Unmanned aircraft collision avoidance using continuous-state POMDPs," in *Robotics: Science and Systems VII*, vol. 1. Cambridge, MA, USA: MIT Press, 2012, pp. 1–8.

[13] S. Nicol and I. Chadès *et al.*, "Which states matter? An application of an intelligent discretization method to solve a continuous POMDP in conservation biology," *PloS One*, vol. 7, no. 2, 2012, Art. no. e28993.

[14] V. Krishnamurthy and D. V. Djonin, "Optimal threshold policies for multivariate POMDPs in radar resource management," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3954–3969, Oct. 2009.

[15] T. Erez and W. D. Smart, "A scalable method for solving high-dimensional continuous POMDPs using local approximation," 2012, *arXiv preprint arXiv:1203.3477*.

[16] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1288–1302, 2014.

[17] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 244–264, 2016.

[18] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Efficient planning in non-Gaussian belief spaces and its application to robot grasping," in *Robotics Research*. New York, NY, USA: Springer, 2017, pp. 253–269.

[19] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst.*, 2014, pp. 392–399.

[20] B. Lavis and T. Furukawa, "HyPE: Hybrid particle-element approach for recursive Bayesian searching and tracking," in *Robotics: Science and Systems IV*. Cambridge, MA, USA: MIT Press, 2009, pp. 135–142.

[21] F. Bourgault, "Decentralized control in a Bayesian world," Ph.D. dissertation, Univ. Sydney, Camperdown, NSW, Australia, 2005.

[22] J. Ousingsawat and M. E. Campbell, "On-line estimation and path planning for multiple vehicles in an uncertain environment," *Int. J. Robust Nonlinear Control*, vol. 14, no. 8, pp. 741–766, 2004.

[23] W. Koch, "On exploiting "negative" sensor evidence for target tracking and sensor data fusion," *Inf. Fusion*, vol. 8, no. 1, pp. 28–39, 2007.

[24] K. Wyffels and M. Campbell, "Negative information for occlusion reasoning in dynamic extended multiobject tracking," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 425–442, Apr. 2015.

[25] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2164–2172.

[26] A. N. Bishop and B. Ristic, "Fusion of spatially referring natural language statements with random set theoretic likelihoods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 932–944, Apr. 2013.

[27] N. Sweet and N. Ahmed, "Structured synthesis and compression of semantic human sensor models for Bayesian estimation," in *Proc. Am. Control Conf.*, 2016, pp. 5479–5485.

[28] T. Kaupp, A. Makarenko, and H. Durrant-Whyte, "Human–robot communication for collaborative decision making: A probabilistic approach," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 444–456, 2010.

[29] K. G. Lore, N. Sweet, K. Kumar, N. Ahmed, and S. Sarkar, "Deep value of information estimators for collaborative human-machine information gathering," in *Proc. 7th Int. Conf. Cyber-Phys. Syst.*, 2016, pp. 3–12.

[30] D. Bonnie, S. Candido, T. Bretl, and S. Hutchinson, "Modelling search with a binary sensor utilizing self-conjugacy of the exponential family," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3975–3982.

[31] A. R. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, Jul. 2007.

[32] D. J. Salmond, "Mixture reduction algorithms for target tracking in clutter," in *Proc. SPIE*, 1990, vol. 1305, p. 434.

[33] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, nos. 1/2, pp. 99–134, 1998.

[34] J. Pineau *et al.*, "Point-based value iteration: An anytime algorithm for POMDPs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, vol. 3, pp. 1025–1032.

[35] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and Systems IV*. Zurich, Switzerland: MIT Press, 2008, pp. 336–343.

[36] M. Rosencrantz, G. Gordon, and S. Thrun, "Locating moving entities in indoor environments with teams of mobile robots," in *Proc. 2nd Int. Joint Conf. Auton. Agents Multi-Agent Syst.*, 2003, pp. 233–240.

[37] M. T. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *J. Artif. Intell. Res.*, vol. 24, pp. 195–220, 2005.

[38] N. Roy and G. J. Gordon, "Exponential family PCA for belief compression in POMDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1667–1674.

[39] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Proc. Mach. Learn.*, 1995, pp. 362–370.

[40] R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search," in *Proc. Int. Conf. Comput. Games*, 2006, pp. 72–83.

[41] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1772–1780.

[42] A. Goldhoorn, A. Garrell, R. Alquézar, and A. Sanfeliu, "Continuous real time POMCP to find-and-follow people by a humanoid service robot," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 741–747.

[43] S. Brechtel, T. Gindele, and R. Dillmann, "Solving continuous POMDPs: Value iteration with incremental learning of an efficient space representation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 370–378.

[44] J. Van Den Berg, S. Patil, and R. Alterovitz, "Efficient approximate value iteration for continuous Gaussian POMDPs," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1832–1838.

[45] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, nos. 11/12, pp. 1448–1465, 2009.

[46] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, "Near-optimal belief space planning via T-LQG," 2017, *arXiv preprint arXiv:1705.09415*.

[47] K. Lesser and M. Oishi, "Approximate safety verification and control of partially observable stochastic hybrid systems," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 81–96, Jan. 2017.

[48] P.-A. Coquelin and R. Munos, "Bandit algorithms for tree search," 2007, *arXiv preprint cs/0703062*.

[49] P. Abbeel, "Reinforcement learning for nonlinear dynamical systems and Gaussian belief space planning," in *Proc. Conf. Reinforcement Learn. Decis. Making*, 2013, p. 13.

[50] J. L. Williams and P. S. Maybeck, "Cost-function-based Gaussian mixture reduction for target tracking," in *Proc. 6th Int. Conf. Inf. Fusion*, 2003, vol. 2, pp. 1047–1054.

[51] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.

[52] N. R. Ahmed, "Data-free/data-sparse softmax parameter estimation with structured class geometries," *IEEE Signal Process. Lett.*, vol. 25, no. 9, pp. 1408–1412, Sep. 2018.

[53] Y. Zhang and H. Ji, "Gaussian mixture reduction based on fuzzy ART for extended target tracking," *Signal Process.*, vol. 97, pp. 232–241, 2014.

[54] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *J. Mach. Learn. Res.*, vol. 18, no. 26, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-300.html

[55] J. R. Schoenberg, M. Campbell, and I. Miller, "Posterior representation with a multi-modal likelihood using the Gaussian sum filter for localization in a known map," *J. Field Robot.*, vol. 29, no. 2, pp. 240–257, 2012.

Luke Burks received the B.S. degree in physics from the University of Arkansas, Fayetteville, AR, USA, in 2015. He is currently working toward the Ph.D. degree in aerospace engineering with the Ann and H.J. Smead Aerospace Engineering Sciences Department, University of Colorado Boulder, Boulder, CO, USA.

His research interests include probabilistic algorithms for autonomous planning under uncertainty, human–robot interaction, and Bayesian data fusion.

Ian Loefgren received the B.S. degree in aerospace engineering from the University of Colorado Boulder, Boulder, CO, USA, in 2018. He received the M.S. degree in aerospace engineering from the Ann and H.J. Smead Aerospace Engineering Sciences Department, University of Colorado Boulder.

His research interests include probabilistic algorithms for decentralized data fusion, and cooperative localization and navigation for autonomous robotics.

Nisar R. Ahmed received the B.S. degree in engineering from Cooper Union, New York City, NY, USA, in 2006 and the Ph.D. degree in mechanical engineering from Cornell University, Ithaca, NY, USA, in 2012.

He is an Assistant Professor of Autonomous Systems with Ann and H.J. Smead Aerospace Engineering Sciences Department, University of Colorado Boulder, Boulder, CO, USA. He was also a Postdoctoral Research Associate with Cornell University until 2014. His research interests include the development of probabilistic models and algorithms for cooperative intelligence in mixed human–machine teams.