Entropy-regularized Point-based Value Iteration

Harrison Delecki[1], Marcell Vazquez-Chanlatte[2], Esen Yel[1] Kyle Wray[1], Tomer Arnon[2],
Stefan Witwicki[2], Mykel J. Kochenderfer[1]

Abstract— **Model-based planners for partially observable problems must accommodate both model uncertainty during planning and goal uncertainty during objective inference. However, model-based planners may be brittle under these types of uncertainty because they rely on an exact model and tend to commit to a single optimal behavior. Inspired by results in the model-free setting, we propose an entropy-regularized model-based planner for partially observable problems. Entropy regularization promotes policy robustness for planning and objective inference by encouraging policies to be no more committed to a single action than necessary. We evaluate the robustness and objective inference performance of entropy-regularized policies in three problem domains. Our results show that entropy-regularized policies outperform non-entropy-regularized baselines in terms of higher expected returns under modeling errors and higher accuracy in objective inference.**

I. INTRODUCTION

Recent applications of partially observable Markov decision processes (POMDPs) have shown success in a range of applications such as autonomous vehicles [1], robotic planning [2], aircraft collision avoidance [3], and objective inference [4]. Model-based planners should be robust to the uncertainty that arises in these different domains. However, modeling errors may cause model-based methods to struggle because they rely on exact knowledge of dynamics and observation models. Additionally, planners are often part of larger decision-making systems that are not included in the model [5]. When inferring another agent's objective, there are often multiple strategies the agent could use to achieve a certain goal. Model-based policies for objective inference are likely to underperform because they overcommit to optimal behaviors [6]. In this work, we propose an entropy-regularizing objective and POMDP planning algorithm to address these issues.

Entropy-regularized objectives are frequently used in model-based planning in a fully observable context [7], model-free reinforcement learning [8], and inverse reinforcement learning (IRL) [6]. In model-free and fully observable planning settings, entropy regularization promotes policy robustness by encouraging the policy to be no more committed to a single optimal action than necessary. In the IRL setting, entropy regularization helps enable inference over multiple, possibly suboptimal strategies for meeting an objective.

Furthermore, entropy regularization produces smoother value functions, which can facilitate optimization [7]. Although previous applications of entropy regularization have found success in fully observable domains, there has been less study of entropy-regularization in the partially observed setting.

Some previous work on regularized entropy planning introduces a reward bonus based on the information entropy of the agent's belief [9] and maximizing the predictability of the agent's behavior [10]. These instances of entropy regularization aim to promote observability in the agent's belief and behavior, respectively. In contrast, our goal is to encourage the planner to consider multiple potential behaviors rather than overcommitting to a single one. A previous approach similar to ours maximizes policy entropy subject to reward-based constraints by synthesizing a finite state controller [11]. Rather than introducing constraints, we introduce a soft regularization term to promote policy entropy. Additionally, we consider belief-based alpha vector policies.

Since the value function of POMDPs is piecewise-linear and convex, point-based value iteration (PBVI) [12] and many other POMDP planning algorithms [13], [14] use alpha vectors to represent approximate value functions and policies. Alpha vector policies are desirable for their auditability and low computational requirements in online applications. Previous work has also investigated entropy-regularized alpha vector policies for the QMDP algorithm, which uses a fully observable approximation of the POMDP [15]. This approximation may cause poor performance, especially when the POMDP involves information gathering actions. We propose an entropy-regularized planning algorithm for POMDPs without the fully observable approximation of QMDP.

In this work, we propose an entropy-regularized objective for model-based planning and an offline point-based solver, Entropy-regularized Point-based Value Iteration (ERPBVI). We evaluate the robustness and objective inference performance of ERPBVI policies in toy problems and an automotive-inspired example. Our experiments show that our entropy-regularized policies are more robust to modeling errors and perform more accurate objective inference than non-entropy-regularized baselines.

II. BACKGROUND

This section provides necessary background on POMDPs, belief-state MDPs, and Point-based Value Iteration.

A. POMDPs

A POMDP is a formulation for sequential decision-making problems where the underlying state is unobservable. POMDPs are represented by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, \gamma \rangle$ where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$ denote the state, action, and observation spaces, respectively. Given a current state $s \in \mathcal{S}$ and agent action $a \in \mathcal{A}$, the agent transitions to a new state s' according to the transition model $s' \sim T\left(\cdot \mid s, a\right)$. The agent cannot observe the true state of the system, but receives an observation $o \sim O\left(\cdot \mid s', a\right)$. The agent uses this observation to update its current belief b over possible states to obtain a posterior distribution $b'(s)$:

$$b'(s') \propto O(o \mid s, a) \sum_{s \in \mathcal{S}} T(s' \mid s, a) b(s)$$

For compactness, we denote the belief update process using the operation $b' = \text{Update}(b, a, o)$.

A stochastic policy $\pi(a \mid b)$ is a distribution over actions given a current belief b. The agent receives a reward specified by $R(s, a)$ for taking action a in state s.

B. Belief-state Markov Decision Process

Any POMDP can be converted into an MDP that uses beliefs as states, also known as a belief-state MDP [16]. In a belief-state MDP, the state space is continuous because it represents the set of all possible beliefs. The action space remains the same as that of the POMDP. For discrete state and action spaces, the belief-state MDP reward is:

$$R(b, a) = \sum_{s \in b} b(s) R(s, a)$$

The belief-state transition model $b' \sim T(\cdot \mid b, a)$ is:

$$
\begin{aligned}
s &\sim b(\cdot) & s' &\sim T(\cdot \mid s, a) \\
o &\sim O(\cdot \mid s', a) & b' &= \text{Update}(b, a, o)
\end{aligned}
$$

C. Point-based Value Iteration

Point-based Value Iteration (PBVI) is an offline approximate solution method for POMDPs [12]. PBVI and other offline solvers represent an approximate value function using a collection of vectors known as alpha vectors. The value function for POMDPs is piecewise linear and convex [17]. PBVI approximates the value function by a set of m vectors $\Gamma = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m\}$. Each alpha vector $\boldsymbol{\alpha}_i$ defines a hyperplane in belief space. The approximate value function is defined by:

$$U^{\Gamma}(\mathbf{b}) = \max_{\boldsymbol{\alpha} \in \Gamma} \boldsymbol{\alpha}^{\top} \mathbf{b}$$

PBVI computes a set Γ of m different alpha vectors associated with m different belief points $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$. The vectors in Γ are initialized to a lower bound on the optimal value function. The algorithm proceeds by alternating between a backup step and an expansion step.

The backup step takes a belief and a set of alpha vectors Γ and constructs a new alpha vector to improve the approximate value function. The backup iterates over every possible
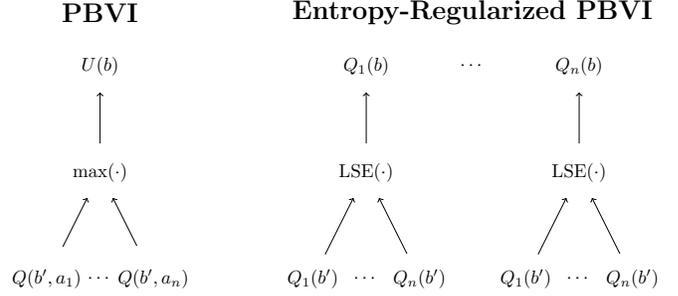

Fig. 1: Illustration of PBVI (left) and entropy-regularized PBVI (right) backup. PBVI updates the estimated value function by maximizing Q-values at next belief points. The entropy-regularized variant explicitly models Q-functions for all n actions, using LogSumExp (LSE) instead of maximum.

action a and observation o, and extracts the maximal alpha vector from the next belief state:

$$\boldsymbol{\alpha}_{a,o} = \arg\max_{\boldsymbol{\alpha} \in \Gamma} \boldsymbol{\alpha}^{\top} \text{Update}(\mathbf{b}, a, o)$$

The algorithm computes a new alpha vector for each available action based on these $\alpha_{a,o}$ vectors, given by:

$$\boldsymbol{\alpha}_a(s) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{o \in \mathcal{O}} O(o \mid s', a) T(s' \mid s, a) \boldsymbol{\alpha}_{a,o}(s')$$

The backed-up alpha vector at belief $\mathbf{b}$ is:

$$\boldsymbol{\alpha} = \arg\max_{\boldsymbol{\alpha}_a} \boldsymbol{\alpha}_a^{\top} \mathbf{b}$$

The expansion step grows the set of belief points B to cover more points in the belief space. The algorithm computes new candidate belief points by computing every action and observation from each point in B. For each belief in B, the expanded belief point that is the furthest away from all other points in B is added to B.

The lower bound on the value function represented by Γ increases with repeated application of the backup until convergence. The converged value function will not be optimal because B typically does not include all beliefs.

III. ENTROPY-REGULARIZED PBVI

Now we introduce an entropy-regularized version of PBVI, which we call ERPBVI. First, we introduce an entropy-regularized objective for planning in a belief-state MDP. Then, we describe details of the ERPBVI algorithm backup step and policy extraction.

A. Entropy-regularized Planning Objective

Here we introduce an entropy-regularized objective in the belief-MDP, inspired by previous work on entropy regularization in the fully-observable setting [18]. The entropy of a stochastic policy π at belief $\mathbf{b}$ is:

$$H(\pi, \mathbf{b}) = -\sum_{a \in \mathcal{A}} \pi(a \mid \mathbf{b}) \log \pi(a \mid \mathbf{b}) \tag{1}$$

We define the entropy-regularized value as

$$U(\mathbf{b}) = \max_{\pi(\cdot|\mathbf{b})\in\mathcal{P}_\mathcal{A}} \mathbb{E}_{a\sim\pi(\cdot|\mathbf{b}),\mathbf{b}'\sim T(\cdot|\mathbf{b},a)}\left[R(\mathbf{b},a)+\gamma U(\mathbf{b}')\right]$$
$$+ \lambda H(\pi,\mathbf{b}) \tag{2}$$

where $\mathcal{P}_\mathcal{A}$ is the set of all probability distributions over $\mathcal{A}$, and $\lambda \geq 0$ and controls the degree of regularization. Since the negative entropy is the convex conjugate of the log-sum-exp function [18], Eq. (2) is equivalent to:

$$U(\mathbf{b}) = \lambda \log\left(\sum_{a\in\mathcal{A}}\exp\left(\frac{1}{\lambda}\mathbb{E}\left[R(\mathbf{b},a)+\gamma U(\mathbf{b}')\right]\right)\right) \tag{3}$$

Since the term in the expectation is equivalent to the Q-value for belief $\mathbf{b}$ and action a, we can write the entropy regularized value function as the log-sum-exp of Q-values weighted by the regularizing strength:

$$U(\mathbf{b}) = \lambda \log\left(\sum_{a\in\mathcal{A}}\exp\left(\frac{1}{\lambda}Q(\mathbf{b},a)\right)\right) \tag{4}$$

For the remainder of this paper, we will use the operator LogSumExp to represent the log-sum-exp similar to Eq. (4).

B. ERPBVI Backup

Rather than computing an approximation of the value function at each belief point, we maintain an estimate of the Q-function for each action. This difference between the PBVI and ERPBVI backup is illustrated in Fig. 1. The Q-function for action a is denoted by the collection of alpha-vectors Γ_a, and the set of all Q-functions is $\Xi = \{\Gamma_1, \ldots, \Gamma_{|\mathcal{A}|}\}$. The set Γ_a defines an approximate Q-function for action a:

$$Q_a(\mathbf{b}) = \max_{\boldsymbol{\alpha}\in\Gamma_a} \boldsymbol{\alpha}^\top \mathbf{b} \tag{5}$$

The value function is then identical to Eq. (4).

The algorithm maintains a lower bound on the optimal value function, $U^\Gamma(\mathbf{b}) \leq U^*(\mathbf{b})$. We initialize each approximate Q-function Γ_a to a lower bound and perform a backup to update the alpha vectors in each Γ_a and point in B. The backup takes in a belief point in B and the set Ξ of Q-functions Γ_a, and constructs a new alpha vector for each Q-function. For each action a, the algorithm iterates through observations o and extracts the alpha vector from each Q-function Γ_a that is maximal at the next belief. The set of dominating alpha vectors is $\Gamma_{a,o}$, where the ith element is:

$$\Gamma_{a,o}^i = \arg\max_{\boldsymbol{\alpha}\in\Gamma_i} \boldsymbol{\alpha}^\top \text{Update}(\mathbf{b},a,o) \tag{6}$$

For convenience, we form the $|\mathcal{A}|$ alpha vectors of $\Gamma_{a,o}$ into the columns of a matrix $\mathbf{A}_{a,o}$. From Eq. (4), we can write the approximate entropy-regularized utility at the next belief-state as:

$$U(\mathbf{b}') = \lambda \text{LogSumExp}\left(\frac{1}{\lambda}\mathbf{A}_{a,o}^\top \text{Update}(\mathbf{b},a,o)\right) \tag{7}$$

Next, we compute an alpha vector representing a linear approximation of the utility at the next belief state. Since

the LogSumExp function is differentiable, we can compute the gradient of Eq. (7) which is an alpha vector $\boldsymbol{\alpha}_{a,o}$ equal to the entropy regularized utility at the next belief:

$$\boldsymbol{\alpha}_{a,o} = \mathbf{A}_{a,o}\,\text{SoftMax}\left(\frac{1}{\lambda}\mathbf{A}_{a,o}^\top \text{Update}(\mathbf{b},a,o)\right) \tag{8}$$

We construct one new alpha vector for each Q-function using all $\boldsymbol{\alpha}_{a,o}$ and a one-step lookahead:

$$\boldsymbol{\alpha}_a(s) = R(s,a)+\gamma\sum_{s',o}O(o\mid a,s')T(s'\mid s,a)\boldsymbol{\alpha}_{a,o}(s') \tag{9}$$

The new alpha vectors are combined with the existing Γ_a. Each approximate Q-function is pruned after every backup to remove dominated alpha vectors. Alpha vector pruning is performed as described by Wray *et al.* [1]. The remainder of the algorithm is the same as PBVI [12]. Pseudocode for ERPBVI is available in Section VI.

C. Policy Extraction

We extract a stochastic policy based on the entropy-regularized objective inspired by previous work in the fully-observable setting [7], [18]. The gradient of the objective in Eq. (4) defines a softmax distribution over actions where the probability of action a_i:

$$\pi(a_i\mid\mathbf{b}) = \frac{\exp(Q_i(\mathbf{b})/\lambda)}{\sum_{j=1}^{|\mathcal{A}|}\exp(Q_j(\mathbf{b})/\lambda)} \tag{10}$$

This is the form of the ERPBVI policy we use in experiments.

IV. EXPERIMENTS

This section discusses experiments designed to evaluate entropy-regularized PBVI policies in robustness and objective inference performance.

A. Robustness Experiments

Robustness experiments evaluate the sensitivity of ERPBVI policies to modeling errors. First, we train a policy on a POMDP problem with known parameters. We evaluate the robustness of policies to environment changes by estimating the expected sum of discounted rewards $\mathbb{E}[R]$ on the same POMDP problem. As a baseline, we consider non-entropy-regularized PBVI policies. Next, we describe POMDP problems used to evaluate robustness.

1) Tiger: We evaluate the robustness of a trained policy to observation noise in the Tiger POMDP by changing the probability of listening correctly for the tiger. The training POMDP uses the observation model as presented in [19]. If the tiger is behind the left door, then the agent will hear the tiger correctly on the left side with probability $p_{correct} = 0.85$. We evaluate the trained policy on problems with $p_{correct} = 0.6$, $p_{correct} = 0.7$ and $p_{correct} = 0.9$. If the agent opens the door and finds the tiger, it gets a reward of -100. If the agent opens the other door, it escapes and receives a reward of 10. The agent receives a reward of -1 for listening. We compute the expected sum of discounted returns using 100 rollouts with horizon 100. For ERPBVI, we train 30 policies with temperature λ varying between 10^{-2} and 10^2 evenly spaced on a log scale.
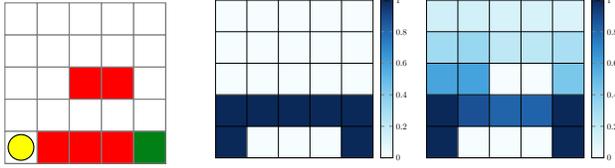
Fig. 2: Illustration of the GridWorld POMDP (left). The agent tries to reach the lower-right corner, avoiding failure states. The expected visitation rate for the PBVI policy (middle) shows that the agent follows a single path to the goal through the narrow passage. The ERPBVI policy (right) follows multiple paths to the goal.

Fig. 3: Illustration of the two GridWorld planning objectives used in the objective inference experiments. The states, actions, and observations are identical in each POMDP.

2) GridWorld: We also consider policy robustness under transition model uncertainty in a GridWorld problem inspired by problems in [20]. States correspond to each location in the grid. The agent can try to move in the four cardinal directions. If the agent hits a wall, the agent stays in the current location. The GridWorld is 'slippery', meaning that there is some probability p_{slip} that the agent will remain in the current location.

When the agent takes an action and the next cell is free, it will slip with probability p_{slip} and remain at the current location. Otherwise, the agent moves to the next location. The grid is configured as shown in Fig. 2. The green cell in the lower right corner represents a goal state, where the agent receives a reward of $+1$ and deterministically transitions to a terminal state. Red squares represent failure states, where the agent receives a -1 reward before deterministically transitioning to the terminal state. The agent can either take a short, risky path or a longer safer path to the goal. There are three observations, corresponding to the three types of grid states (normal, goal, and failure). The agent always starts in the bottom left corner.

We train policies with $p_{slip} = 0$, and evaluate robustness on $p_{slip} = 0.1$, $p_{slip} = 0.3$, and $p_{slip} = 0.5$. We evaluate the robustness of ERPBVI policies with 30 temperatures between 10^{-4} and 10^{2} evenly spaced on a log scale.

B. Objective Inference Experiments

The objective inference experiments are inspired by previous work in IRL and goal recognition [4], [6]. Suppose we observe an agent acting in a POMDP with a set $\mathcal{G}$ of possible goals G. For example, there may be multiple goal locations in a simple navigation domain. We assume that we observe the agent's action-observation sequence $\tau = [a_1, o_1, \ldots, a_t, o_t]$ up to time step t. We also assume that the agent's initial belief $\mathbf{b}_0$ is known, so that we can compute the agent's belief state at any step.

Our goal is to compute the posterior probabilities $P(G \mid \tau)$. Assuming a uniform prior over goals, the inference query $P(G \mid \tau)$ can be expressed:

$$P(G \mid \tau) \propto P(\tau \mid G) \qquad (11)$$

A key assumption to approximate $P(\tau \mid G)$ is that if the agent is pursuing goal G, then the agent will choose action a in belief-state $\mathbf{b}$ with probability

$$P(a \mid \mathbf{b}, G) = \pi_G(a \mid \mathbf{b}) \qquad (12)$$

where π_G is a stochastic policy trained to maximize the objective for goal G. We can approximate the probability of the action-observation sequence τ given goal G as

$$P(\tau \mid G) \approx \prod_{i=1}^{t} \pi_G(a_i \mid \mathbf{b}_i) \qquad (13)$$

where $\mathbf{b}_i$ is the belief-state computed from τ for time step i. After computing the posterior $P(G \mid \tau)$ for each goal, we take the maximum likelihood estimate to predict G.

To evaluate the objective inference performance, we compare the true-positive rate, false-positive rate, and accuracy of using ERPBVI and PBVI policies. We train ERPBVI policies for each goal G, and use the stochastic policy from Eq. (10) to compute action probabilities. As a baseline, we build a stochastic policy from a softmax distribution over non-entropy regularized PBVI Q-values

$$\pi^{PBVI}(a \mid \mathbf{b}) \propto \exp(Q(\mathbf{b}, a)/\lambda) \qquad (14)$$

where $Q(\mathbf{b}, a)$ is the action-value using the PBVI alpha vectors and λ is a temperature parameter. Next, we describe the GridWorld and Crosswalk POMDPs used in objective inference experiments.

1) GridWorld: We perform objective inference in a Grid-World POMDP identical to the one presented in robustness experiments but without the failure cells. The two objectives, visualized in Fig. 3, each have a single goal location against one of the far walls from the agent's initial position. We generate 1000 action-observation trajectories using the deterministic PBVI policies that select random actions with probability 0.5. Each method observes the first 5 time steps of each trajectory. We compare each method's true-positive rate and false-positive rate for 10 temperatures between 10^{-2} and 10^{2} spaced linear on a log scale.

2) Crosswalk: We model a pedestrian crossing the street at a crosswalk with a single oncoming car. The problem's state is a tuple $\langle p_{ped}, p_{car}, v_{car} \rangle$ where p_{ped} is the position of the pedestrian along the crosswalk, p_{car} is the position of the vehicle in the street perpendicular to the crosswalk, and v_{car} is the vehicle's velocity. The pedestrian's actions include moving forward, moving backward, or remaining stationary. The crosswalk and adjoining street are discretized into 10 positions. The oncoming vehicle can be moving forward zero, one or two units per time step. The vehicle's dynamics depend solely on whether the pedestrian is in the street. When the pedestrian is on the side of the road, the oncoming
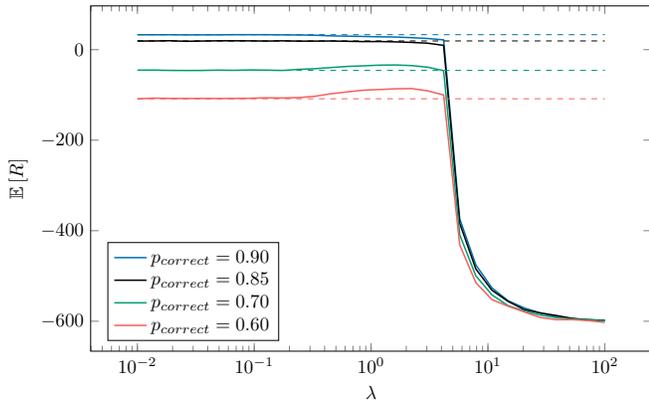
Fig. 4: Robustness results on Tiger POMDP using ERPBVI (solid line) and PBVI (dashed line) for various $p_{correct}$. The nominal $p_{correct} = 0.85$ is shown in black. When the model is worse than expected ($p_{correct} \leq 0.85$), ERPBVI achieves higher expected return $\mathbb{E}[R]$ than PBVI for $\lambda \approx 1$.

Fig. 5: Expected return for PBVI and ERPBVI policies in Grid-World with varying p_{slip}. All policies are trained with $p_{slip} = 0$. Solid lines indicate ERPBVI performance, while dashed lines indicate PBVI. ERPBVI is less sensitive to changes in the POMDP's transition model when $\lambda \approx 0.02$.

TABLE I: Maximum increase in mean return Δ for ERPBVI versus PBVI on Tiger and GridWorld.

| | Tiger | | | GridWorld | | |
| | $p_{correct}$ | | | p_{slip} | | |
	0.9	0.7	0.6	0.1	0.3	0.5
Δ	0.0	11.81	22.62	0.03	0.09	0.06

vehicle brakes with probability $p_b = 0.5$. If the pedestrian is in the street, the vehicle brakes with a larger probability $p_c = 0.9$. The street begins at position 3 along the crosswalk. If the pedestrian is in the street $p_{ped} > 3$ and $p_{car} \in [7, 9]$, there is a collision. The agent receives noisy observations of the vehicle's position.

We use two variants of this problem for objective inference. In the first version, the pedestrian incurs a large penalty for being struck by the vehicle. This penalty prioritizes safe street-crossing behaviors, emphasizing avoidance of the oncoming car. In contrast, the second version features no penalty when the pedestrian is struck, representing a scenario where the pedestrian disregards oncoming traffic. We use the same data generation parameters as in the GridWorld experiment. Each method observes the first 10 time steps of each trajectory. We compare each method's true-positive rate and false-positive rate for nine temperatures between 10^{-4} and 10^4 linearly spaced on a logarithmic scale.

V. RESULTS

This section presents results of the robustness and objective inference experiments using ERPBVI.

A. Robustness

First, we present results of the robustness experiments on the Tiger and Gridworld POMDPs. The maximum improvement in expected return using ERPBVI compared to PBVI is shown in Table I.

1) Tiger: Robustness results for the Tiger POMDP are shown in Fig. 4. The solid line indicates ERPBVI performance, and the dashed line indicates the regular PBVI performance. The black lines show performance on the training problem. ERPBVI can only perform as well as PBVI when the evaluation problem is identical to the training problem. As the policy temperature is increased, the ERPBVI policy converges to the uniform random policy, leading to much worse performance.
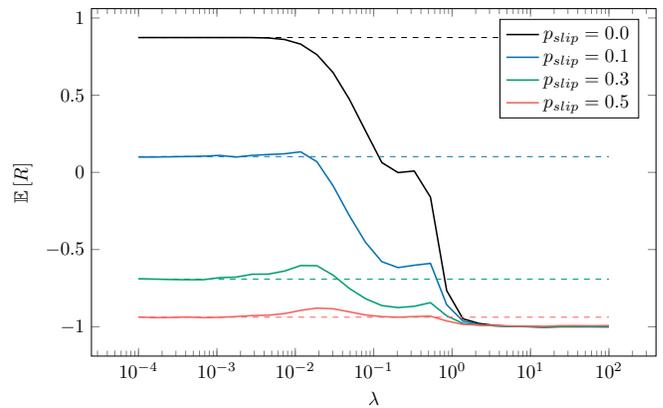
When the true $p_{correct}$ is smaller than during training, there exists a range of temperatures where ERPBVI outperforms PBVI. However, when the observations are less noisy than training, the entropy-regularized policies cannot perform any better. This is because entropy regularization leads to policies that place less trust in the observation model.

2) GridWorld: Robustness results for the GridWorld problem are shown in Fig. 5. As before, the solid line denotes ERPBVI and the dashed line denotes PBVI. For certain temperature ranges, the entropy-regularized policy achieves a greater expected discounted return than the PBVI policy. As illustrated in Fig. 2, the PBVI policy always attempts to cross the narrow corridor to the goal, as it is trained in a deterministic environment. In contrast, the stochastic entropy-regularized policy takes both the short, risky path and the safer path around the failure states.

B. Objective Inference

Here we present the results of the inference experiments on the GridWorld and Crosswalk POMDPs.

1) GridWorld: Results for objective inference in the Grid-World are shown in Fig. 6. The plot shows the true-positive rate versus false-positive rate using ERPBVI (squares) and PBVI baseline (circles) policies with a variety of temperatures. The dashed line indicates the theoretical performance of a random classifier. Assuming equal preference between true positives and false positives, the ideal performance would be in the upper-left corner. At low temperatures, ERPBVI achieves better performance than the PBVI baseline with higher true-positive rates and lower false-positive rates. Inference using ERPBVI policies is less sensitive to
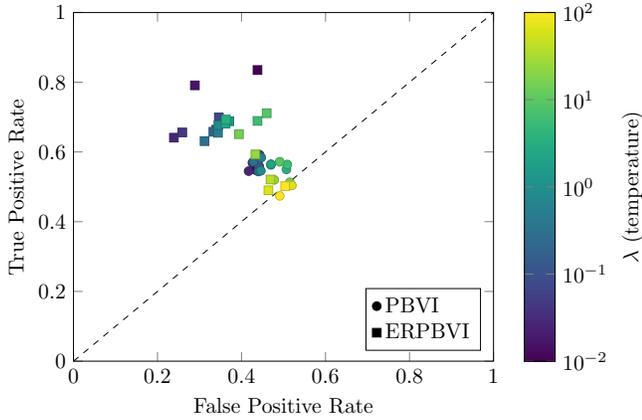
Fig. 6: True positive rate versus false positive rate for ERPBVI and PBVI policies on the GridWorld POMDP. For low temperatures, ERPBVI policies perform better (higher TPR and lower FPR) than the PBVI policies. When the temperature is large, the inference performance for both methods decays to random guessing.
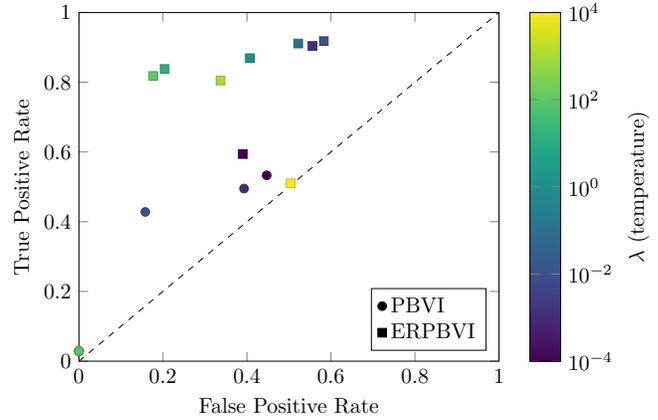

Fig. 7: True positive rate versus false positive rate for ERPBVI and PBVI policies on the Crosswalk POMDP. The positive classification corresponds to the safe behavior of avoiding collision. ERPBVI policies significantly outperform PBVI policies with higher true positive rate and lower false positive rate. Six PBVI points overlap at the lower left.

random or sub-optimal action sequences because the entropy-regularization encourages the policy to be less committed to a single optimal action. When the temperature increases, the inference performance of both ERPBVI and PBVI reverts to random guessing, since the high-temperature softmax induces a uniform distribution over actions.

2) Crosswalk: True positive rate versus false positive rate for the Crosswalk problem is shown in Fig. 7. ERPBVI significantly outperforms PBVI for some temperature ranges with higher true positive rates and lower false positive rates. The difference in performance is more dramatic in this problem because PBVI overcommits to a single optimal behavior of the pedestrian. For example, the optimal policy for the pedestrian following the unsafe objective is to always go forward. If a pedestrian following this objective were to stutter, PBVI would assign a very low probability to that objective. In contrast, ERPBVI can better account for multiple potential paths under the same objective.

VI. CONCLUSION

Model-based POMDP planners must be robust to uncertainty in planning and objective inference. In this work, we present ERPBVI, an entropy-regularized point-based solver for POMDPs. Our experiments on toy and automotive-inspired POMDPs show that ERPBVI is more robust to modeling errors than PBVI, and also outperforms PBVI in objective inference. In future work, we plan to investigate the theoretical properties of the ERPBVI algorithm. Furthermore, we will investigate how entropy-regularized POMDP policies can be used to solve decomposed POMDPs [5], [21]. Since entropy-regularized policies are less committed to single optimal actions, they may be useful for fusing multiple utility estimates from decomposed POMDPs.

APPENDIX

This section presents pseudocode for the ERPBVI algorithm, which shares much of its code with PBVI. The high-level algorithm shown in Algorithm 1 takes in a POMDP

formulation $\mathcal{P}$ and regularization temperature λ. The belief point set $\mathcal{B}$ is initialized with the POMDP agent's initial belief-state, and Q-functions are initialized to a lower bound.

ERPBVI iteratively improves Q-function estimates, prunes unnecessary alpha-vectors, and expands the belief set over a fixed number of iterations. The improvement step backs up the Q-function at each point in $\mathcal{B}$. The backup procedure is shown in Algorithm 2. Several backups may be needed for convergence to a new backed-up alpha vector for each Q-function. After backup, Q-functions are pruned by solving a linear program to remove dominated alpha vectors [16]. The expansion step expands every possible next belief from each belief in $\mathcal{B}$. The farthest points from points in $\mathcal{B}$ are added to $\mathcal{B}$. An open source implementation of ERPBVI is available[1].

Algorithm 1 Entropy-regularized Point-based Value Iteration

Require: $\mathcal{P} \overset{\text{def}}{=} \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, O, \gamma \rangle$: POMDP
Require: λ: Temperature
1: **function** ERPBVI($\mathcal{P}, \lambda$)
2:     $B \leftarrow$ INITIALIZEBELIEFS($\mathcal{P}$)
3:     $\Xi \leftarrow$ INITIALIZEQVALUES($\mathcal{P}$)
4:     **for** $i \leftarrow 1$ **to** $n_{\text{iterations}}$
5:        **for** $\mathbf{b}$ in B ▷ Q-function improvement
6:           $\Xi \leftarrow$ BACKUP($\mathcal{P}, \mathbf{b}, \Xi$)
7:        $\Xi \leftarrow$ PRUNE(Ξ)   ▷ Q-value pruning
8:        $B \leftarrow$ EXPAND($\mathcal{P}, B$)   ▷ Belief expansion
9:     **return** $\pi(\Xi)$ ▷ Alpha-vector policy

Algorithm 2 Backup

1: **function** BACKUP($\mathcal{P}, \mathbf{b}, \Xi$)
2:     **for** a in $\mathcal{A}$
3:        **for** o in $\mathcal{O}$
4:           $\mathbf{b}' \leftarrow$ UPDATE($\mathbf{b}, a, o$)
5:           **for** Γ_i in Ξ
6:              $A_{a,o}(i) \leftarrow$ ARGMAX($\Gamma_i, \mathbf{b}'$)   ▷ Eq. (6)
7:           $\alpha_{a,o} \leftarrow A_{a,o}$SOFTMAX($A_{a,o}^{\top}\mathbf{b}'/\lambda$)   ▷ Eq. (8)
8:        $\alpha_a \leftarrow$ LOOKAHEAD($\mathcal{P}, \alpha_{a,o}$)   ▷ Eq. (9)
9:        APPEND(Ξ, α_a)
10:     **return** Ξ

---

[1]https://github.com/hdelecki/EntropyRegularizedPBVI.jl

REFERENCES

[1] K. H. Wray, B. Lange, A. Jamgochian, S. J. Witwicki, A. Kobashi, S. Hagaribommanahalli, and D. Ilstrup, "POMDPs for safe visibility reasoning in autonomous vehicles," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, 2021, pp. 191–195.

[2] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable Markov decision processes in robotics: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 21–40, 2022.

[3] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, "Next generation airborne collision avoidance system," *Lincoln Laboratory Journal*, vol. 19, no. 1, pp. 17–33, 2012.

[4] M. Ramırez and H. Geffner, "Goal recognition over POMDPs: Inferring the intention of a POMDP agent," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011, pp. 2009–2014.

[5] K. H. Wray, S. J. Witwicki, and S. Zilberstein, "Online decision-making for scalable autonomous systems," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, 4768–4774.

[6] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2008, pp. 1433–1438.

[7] J.-B. Grill, O. Darwiche Domingues, P. Ménard, R. Munos, and M. Valko, "Planning in entropy-regularized Markov decision processes and games," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning (ICML)*, 2018, pp. 1861–1870.

[9] A. Curtis, L. Kaelbling, and S. Jain, "Task-directed exploration in continuous POMDPs for robotic manipulation of articulated objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3721–3728.

[10] T. L. Molloy and G. N. Nair, "Entropy-regularized partially observed Markov decision processes," *IEEE Transactions on Automatic Control*, 2023.

[11] Y. Savas, M. Hibbard, B. Wu, T. Tanaka, and U. Topcu, "Entropy maximization for partially observable Markov decision processes," *IEEE Transactions on Automatic Control*, vol. 67, pp. 6948–6955, 2022.

[12] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003, pp. 1025–1030.

[13] M. T. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[14] 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*, 2008.

[15] M. Park, J. Shin, and I. Yang, "Anderson acceleration for partially observable Markov decision processes: A maximum entropy approach," *Automatica*, vol. 163, p. 111 557, 2024.

[16] M. J. Kochenderfer, T. A. Wheeler, and K. H. Wray, *Algorithms for Decision Making*. MIT Press, 2022.

[17] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[18] M. Geist, B. Scherrer, and O. Pietquin, "A theory of regularized Markov decision processes," in *International Conference on Machine Learning (ICML)*, 2019, pp. 2160–2169.

[19] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman, "Acting optimally in partially observable stochastic domains," in *AAAI Conference on Artificial Intelligence (AAAI)*, 1994, pp. 1023–1028.

[20] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning (ICML)*, 1995, pp. 362–370.

[21] M. Bouton, K. D. Julian, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Decomposition methods with deep corrections for reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, pp. 330–352, 2019.