2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160455

Learning Continuous Control Policies for Information-Theoretic Active Perception

Pengzhi Yang Yuhan Liu Shumon Koga Arash Asgharivaskasi Nikolay Atanasov

***Abstract*— This paper proposes a method for learning continuous control policies for exploration and active landmark localization. We consider a mobile robot detecting landmarks within a limited sensing range, and tackle the problem of learning a control policy that maximizes the mutual information between the landmark states and the sensor observations. We employ a Kalman filter to convert the partially observable problem in the landmark states to a Markov decision process (MDP), a differentiable field of view to shape the reward function, and an attention-based neural network to represent the control policy. The approach is combined with active volumetric mapping to promote environment exploration in addition to landmark localization. The performance is demonstrated in several simulated landmark localization tasks in comparison with benchmark methods.**

I. Introduction

Recent advances in embedded sensing and computation hardware and in simultaneous localization and mapping (SLAM) software have enabled efficient, reliable, real-time mapping of unknown and unstructured environments [1]. However, most robot mapping methods are passive in utilizing sensing information and do not consider optimizing the robot's motion to improve performance. Yet, planning the robot's sensing trajectory to improve the quality of acquired information [2] may play a critical role in challenging environments in applications such as search and rescue [3], security and surveillance [4], and wildfire detection [5].

This paper proposes an approach to learn continuous control policies for active perception. We consider a robot equipped with an onboard sensor capable of detecting objects of interest (landmarks) within a limited field of view (FoV). The objective is to maximize the mutual information between the landmark states and potential future sensor observations given past sensory data and robot trajectory. With a sensor model that is linear in the landmark states and subject to Gaussian noise, the mutual information objective is related to the information matrix of a Kalman filter (KF) estimating the landmark states. To prevent a non-smooth reward function due to the limited FoV, we use a differentiable FoV formulation for reward shaping. Then, an exploration policy is learned using proximal policy optimization (PPO) [6] over a continuous control space with a network architecture using an attention mechanism to handle multiple landmarks. The proposed method is demonstrated in simulation in comparison to an open-loop optimization method and a policy with a different network architecture.

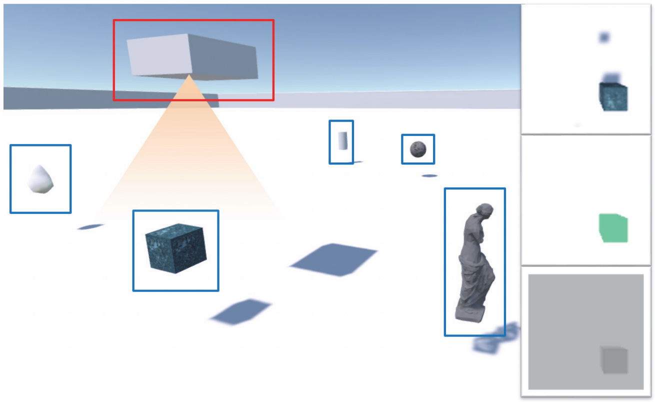

Fig. 1: Active landmark localization in the Unity simulator [7]. The left figure shows a third-person view of an agent (red box) with limited field of view (orange triangle) tasked with exploring and localizing landmarks (blue boxes) in the environment. The images on the right from top to bottom are RGB, semantic segmentation, and depth images obtained by the agent's onboard sensor.

Frontier-based exploration [8] is one of the early techniques for autonomous robot exploration. It drives the robot to map frontiers, separating explored and unexplored space. To accommodate sensing noise and uncertainty, information-based planning has been explored in many recent works including [9]–[13]. While many information-based exploration techniques assume a discrete control space composed of a finite number of action choices, [14] proposed a gradient ascent approach, named iterative Covariance Regulation (iCR), which optimizes an information objective over continuous control space. The authors employed a differentiable FoV to obtain a differentiable objective. Because iCR provides an open-loop control sequence for a given environment, the solution cannot be applied to a new environment without online re-planning.

Learning a control policy from data is a central problem in reinforcement learning (RL) [15]. RL techniques coupled with deep learning representations have had impressive success in games [16], where the action space is often discrete. More recently, deep RL algorithms have been developed for continuous control [6], [17], which is necessary for various robotics tasks, including visual navigation for mobile [18] and humanoid [19] robots. In this paper, we consider learning a control policy for active perception as an alternative to view planning at execution time. Closely related to our work, Jeong et al. [20] applied Q-learning to an active target track-

We gratefully acknowledge support from NSF FRR CAREER 2045945.

The authors are with the Department of Electrical and Computer Engineering, UC San Diego, 9500 Gilman Drive, La Jolla, CA, 92093, USA, {peyang,yul139,skoga,aasghari,natanasov}@ucsd.edu.

An open-source implementation is available at https://github.com/JaySparrow/RL-for-active-mapping.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.

ing problem to maximize the mutual information between sensor data and the target states. Hsu et al. [21] developed a multi-agent version of [20] by incorporating an attention-block in the Q-network architecture. Chen et al. [22] focused on learning exploration policies in a sample-efficient way by applying imitation learning first and then employing PPO for further improvement using a coverage reward. Julian and Kochenderfer [5] used deep Q-learning for belief-space planning in a distributed wildfire surveillance scenario, modelled as a Partial Observable Markov Decision Process (POMDP). Chen et al. [23] tackled active exploration for landmark mapping using a graph neural network representing the exploration policy trained with deep Q learning and advantage actor critic methods. Chaplot et al. [24] propose a modular and hierarchical approach to obtain a local policy by imitation learning from analytical path planners with a learned SLAM module and a global policy to maximize area coverage. Our work differs from [24] since we consider an information-theoretic objective with continuous control that does not rely on learning from any prior geometric planner. Lodel et al. [25] propose an information-theoretic objective to maximize conditional mutual information between a map estimate and new observations given past sensor data. The authors apply PPO to acquire reference view points that maximize the reward with local sensing of obstacles and the robot position. Our work differs from [25] in that we incorporate the posterior mean of the Kalman filter in the state and we use neural network attention to handle multiple landmarks.

The **contribution** of this paper is an approach for learning continuous control policies for active perception with information-theoretic reward, employing a differentiable FoV and an attention-based policy architecture. Our evaluation demonstrates that our method outperforms control policies with different neural network architectures and pre-computed exploration trajectories in a landmark localization problem and can also be utilized for simultaneous exploration of an unknown environment.

II. Problem Statement

This section formalizes active exploration and mapping as an optimal control problem. Consider a robot with state $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and control input $\mathbf{u}_k \in \mathbb{R}^{n_u}$ at discrete time $k \in \mathbb{Z}_{\geq 0}$. The objective is to plan a trajectory to localize several landmarks $\mathbf{y} = [\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n_l)}]$, where $\mathbf{y}^{(j)} \in \mathbb{R}^2$ for $j \in \{1, \ldots, n_l\}$ denotes the position of j-th landmark and n_l is the total number of landmarks. We model the motion of the robot using deterministic nonlinear dynamics:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \tag{1}$$

where $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is a given function.

Let $\mathcal{F} \subset \mathbb{R}^2$ represent the FoV of the onboard sensors within the robot's body frame. The set of landmark indices within the FoV is:

$$\mathcal{I}_{\mathcal{F}}(\mathbf{x}, \{\mathbf{y}^{(j)}\}) = \left\{ j \in \{1, \ldots, n_l\} \mid \mathbf{q}\left(\mathbf{x}, \mathbf{y}^{(j)}\right) \in \mathcal{F} \right\} \tag{2}$$

where $\mathbf{q}(\mathbf{x}, \mathbf{y}^{(j)})$ returns the robot-body-frame coordinates of $\mathbf{y}^{(j)}$. Then, a sensor measurement is denoted by $\mathbf{z}_k = [\{\mathbf{z}_k^{(j)}\}_{j \in \mathcal{I}_{\mathcal{F}}(\mathbf{x}_k, \{\mathbf{y}^{(j)}\})}] \in \mathbb{R}^{n_z |\mathcal{I}_{\mathcal{F}}(\mathbf{x}_k, \{\mathbf{y}^{(j)}\})|}$ where $\mathbf{z}_k^{(j)} \in \mathbb{R}^{n_z}$ is an observation of j-th landmark with model:

$$\mathbf{z}_k^{(j)} = H(\mathbf{x}_k)\mathbf{y}^{(j)} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, V(\mathbf{x}_k)), \tag{3}$$

for all $j \in \mathcal{I}_{\mathcal{F}}(\mathbf{x}_k, \{\mathbf{y}^{(j)}\})$, where the matrix $H(\mathbf{x}) \in \mathbb{R}^{n_z \times 2}$ captures the dependence of the observation on the robot state $\mathbf{x}$ and $V(\mathbf{x}) \in \mathbb{R}^{n_z \times n_z}$ is the sensing noise covariance.

We aim to maximize the conditional mutual information between the landmark states $\mathbf{y}$ and a new observation $\mathbf{z}_{k+1}$ conditioned on the past observations and robot states, denoted as $I(\mathbf{y}; \mathbf{z}_{k+1} | \mathbf{z}_{1:k}, \mathbf{x}_{1:k+1})$. Due to the Gaussian sensor model (3), the conditional mutual information is given by:

$$I(\mathbf{y}; \mathbf{z}_{k+1} | \mathbf{z}_{1:k}, \mathbf{x}_{1:k+1}) = \frac{1}{2} \left(\log\det(Y_{k+1}) - \log\det(Y_k)\right) \tag{4}$$

where $Y_k \in \mathbb{S}_{\succ 0}^{2n_l \times 2n_l}$ is a symmetric positive-definite information matrix obtained by Kalman filter updates. Since the landmark measurements are independent, the information matrix is block-diagonal $Y_k = \operatorname{diag}(Y_k^{(1)}, \ldots, Y_k^{(n_l)})$ with $Y_k^{(j)} \in \mathbb{S}_{\succ 0}^{2 \times 2}$. Due to the limited FoV sensor model, the update of the information matrix is applied only to the indices j within the set $\mathcal{I}_{\mathcal{F}}(\mathbf{x}, \{\mathbf{y}^{(j)}\})$:

$$Y_{k+1}^{(j)} = Y_k^{(j)} + M(\mathbf{x}_{k+1}), \tag{5}$$

$$M(\mathbf{x}) := H(\mathbf{x})^\top V(\mathbf{x})^{-1} H(\mathbf{x}). \tag{6}$$

Summarizing the formulation above, the active perception problem we address in this paper is presented below.

Problem. Obtain a control policy π that solves the infinite-horizon stochastic optimal control problem:

$$\max_{\pi} \mathbb{E}_\pi \left(\sum_{k=0}^{\infty} \gamma^k \left(\log\det\left(Y_{k+1}\right) - \log\det(Y_k)\right) \right), \tag{7}$$

subject to (1)–(3) and (5)–(6), where the function π maps a robot state $\mathbf{x}_k$ and measurement sequence $\mathbf{z}_{0:k}$ to a control input $\mathbf{u}_k$ and $\gamma \in [0, 1)$ is a discount factor.

III. Continuous Control for Active Perception

This section presents our method for learning a continuous control policy to reduce the uncertainty in the landmark states. An overview of the method is shown in Fig. 2.

A. MDP Formulation with Kalman Filter

The problem stated in the previous section is a POMDP [26] since the landmark states are unobservable and the control policy should take the history of sensory data $\mathbf{z}_{1:k}$ into account. To avoid the POMDP complexity, we convert the problem into an equivalent MDP, by using a Kalman filter to compute a sufficient statistic of the state $\mathbf{y}_k$ given the observation history $\mathbf{z}_{0:k}$, namely the posterior distribution $\mathbf{y}_k | \mathbf{z}_{1:k} \sim \mathcal{N}\left(\boldsymbol{\mu}_k, Y_k^{-1}\right)$. The posterior mean $\boldsymbol{\mu}_k \in \mathbb{R}^{2n_l}$ and covariance (inverse of the information matrix) Y_k^{-1} are updated every time a new measurement is obtained. This setting is formalized as an MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, with continuous state space $\mathcal{S}$, continuous action space $\mathcal{A}$, transition

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.

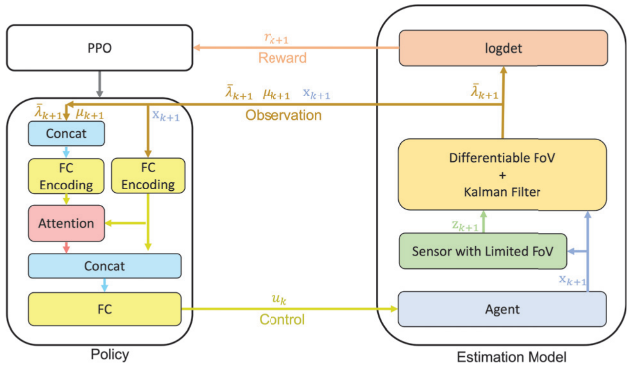

Fig. 2: Estimation and control architecture used for active perception. A Kalman filter (right) updates the mean vector $\boldsymbol{\mu}_k$ and vectorized information matrix $\boldsymbol{\lambda}_k$ of several landmarks of interest using a sensor observation $\mathbf{z}_{k+1}$ obtained from agent position $\mathbf{x}_{k+1}$. A reward function based on the $\log\det$ of the information matrix obtained with a differentiable field of view and the proximal policy optimization algorithm [6] are used to optimize the parameters of a control policy, whose representation (left) uses an attention mechanism to capture the relationship between the agent's position and the landmark states.

probability distribution $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, stage reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. To define the MDP state variable, let $\boldsymbol{\lambda}_k := \text{vech}(Y_k) \in \mathbb{R}^{n_\lambda}$ be the half-vectorization of the symmetric information matrix Y_k, which stacks the columns of the lower-triangular part of Y_k into a vector with dimension $n_\lambda := n_l(2n_l + 1)$. Then, denote the Kalman filter update from time k to $k + 1$ for $j \in \mathcal{I}_\mathcal{F}(\mathbf{x}, \{\mathbf{y}^{(j)}\})$ as:

$$[\boldsymbol{\mu}_{k+1}^{(j)}, \boldsymbol{\lambda}_{k+1}^{(j)}] = KF(\boldsymbol{\mu}_k^{(j)}, \boldsymbol{\lambda}_k^{(j)}, \mathbf{z}_{k+1}, \mathbf{x}_{k+1}). \tag{8}$$

We describe the design of the stage reward function r next.

B. Reward Shaping by Differentiable FoV

Typical RL methods for continuous control employ policy gradient methods [17]. However, due to the limited sensor FoV in (2), the information update in (5) may cause a sudden jump in the stage reward in (7) with respect to the robot state and action. Such non-smooth behavior in the reward may prevent a policy gradient method from converging to a desired solution. To design a smooth reward function, we apply a differentiable FoV formulation [14], which serves as a reward shaping. Namely, the information update (5) is performed for all $j \in \{1, \ldots, n_l\}$, where the matrix $M(\mathbf{x})$ in (6) is replaced as follows:

$$M(\mathbf{x}, \boldsymbol{\mu}^{(j)}) = \left(1 - \Phi(d(\mathbf{q}(\mathbf{x}, \boldsymbol{\mu}^{(j)}), \mathcal{F}))\right) M(\mathbf{x}), \tag{9}$$

where Φ is a probit function [27], defined by the Gaussian CDF $\Phi : \mathbb{R} \to [0,\ 1]$, $\Phi(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2\kappa}} - 2\right)\right]$, where $\text{erf}(y) := \frac{2}{\sqrt{\pi}} \int_0^y e^{-t^2} dt$, and d is a signed distance function associated with the FoV $\mathcal{F}$ defined below.

Definition 1. The *signed distance function* $d : \mathbb{R}^2 \to \mathbb{R}$ associated with a set $\mathcal{F} \subset \mathbb{R}^2$ is:

$$d(\mathbf{q}, \mathcal{F}) = \begin{cases} -\min_{\mathbf{q}^* \in \partial\mathcal{F}} ||\mathbf{q} - \mathbf{q}^*||, & \text{if} \quad \mathbf{q} \in \mathcal{F}, \\ \ \ \min_{\mathbf{q}^* \in \partial\mathcal{F}} ||\mathbf{q} - \mathbf{q}^*||, & \text{if} \quad \mathbf{q} \notin \mathcal{F}, \end{cases} \tag{10}$$

where $\partial\mathcal{F}$ is the boundary of $\mathcal{F}$.

With the proposed differentiable FoV, the information vector $\boldsymbol{\lambda}_k$ in the Kalman Filter is replaced with the new variable $\bar{\boldsymbol{\lambda}}_k$ in the observation space. Overall, the MDP state vector $\mathbf{s}_k \in \mathbb{R}^{n_x + 2n_l + n_\lambda}$ is defined as:

$$\mathbf{s}_k = \begin{bmatrix} \mathbf{x}_k^\top & \bar{\boldsymbol{\lambda}}_k^\top & \boldsymbol{\mu}_k^\top \end{bmatrix}^\top \tag{11}$$

and the action is $\mathbf{a}_k = \mathbf{u}_k$. The robot motion model in (1) and the KF update in (8) together with the differentiable FoV in (9) define the MDP transition model.

C. PPO with Attention-Based Network Architecture

The return from a state is defined as the sum of discounted future rewards, $R_k = \sum_{j=k}^{\infty} \gamma^{j-k} r(\mathbf{s}_j, \mathbf{a}_j)$. The value function $V^\pi : \mathcal{S} \to \mathbb{R}$, the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and the advantage function $A^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ under a control policy π are defined by $V^\pi(\mathbf{s}) = \mathbb{E}_\pi\left[R_0 | \mathbf{s}_0 = \mathbf{s}\right]$, $Q^\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi\left[R_0 | \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}\right]$, $A^\pi(\mathbf{s}, \mathbf{a}) = Q^\pi(\mathbf{s}, \mathbf{a}) - V^\pi(\mathbf{s})$. We apply PPO [6], an actor-critic method for learning the neural network models of both the value function and the control policy. PPO has shown superior performance for continuous control tasks with smooth policy updates. The policy π_θ is updated to minimize a clipped surrogate objective:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}\left[\min\left\{\frac{\pi_\theta}{\pi_{\theta_k}}\hat{A}_k, \text{clip}\left(\frac{\pi_\theta}{\pi_{\theta_k}}, 1 - \epsilon, 1 + \epsilon\right)\hat{A}_k\right\}\right], \tag{12}$$

where $\hat{A}_k$ is an estimate of the advantage function, set as $\hat{A}_k = \sum_{i=0}^{T-k+1} (\gamma\lambda')^i \, \delta_{k+i}$ where $\delta_k = r_k + \gamma V_\phi(s_{k+1}) - V_\phi(s_k)$, and λ' represents the parameter controlling the balance between value estimations' bias and variance. The value function is subsequently updated to minimize $\sum_{k=0}^{T}(V_\phi(\mathbf{s}_k) - R_k)^2$.

To take a reasonable trajectory that maximizes information gain, the agent should plan based on its previous observations. Some related works directly incorporate historical observations and agent states or use recurrent neural networks [23], [25]. In contrast, our model uses the landmark means and information matrices as part of the state, and because they form a sufficient statistic it is not necessary to have historical information.

Another important aspect of our design is that the agent should be aware of the relationship between its current position and the landmark states in order to prioritize observing uncertain landmarks. To capture the position relationship we employ an attention mechanism [28] in the design of the policy architecture. We encode the agent position $\mathbf{x}_k \in \mathbb{R}^{n_x}$ into $\text{Emb}(\mathbf{x})$ with two 32-unit fully connected layers (FC), and the landmark states $\mathbf{o}_k^{\text{lm}} \in \mathbb{R}^{n_l \times 4}$ (reshaped from $[\boldsymbol{\lambda}_k^\top \ \boldsymbol{\mu}_k^\top]^\top \in \mathbb{R}^{4n_l}$) into $\text{Emb}(\mathbf{o}_k^{\text{lm}})$ with a 64-unit FC followed by a 32-unit FC. Then, we obtain the agent-landmark relationship embedding denoted as Emb_{RELP} by calculating the attention of $\text{Emb}(x)$ over all $\text{Emb}(o_k^{\text{lm}})$. Finally, we concatenate $\text{Emb}(x)$ and Emb_{RELP} and pass through two 64-unit fully connected layers to compute the action or value. An illustration of our model is shown in Fig. 2.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.

D. Joint Exploration and Landmark Localization

In addition to landmark localization, we consider simultaneous exploration of an unknown environment. A map state $\mathbf{m} \in \mathbb{R}^{n_m}$ is defined by discretizing the space $\Omega \subset \mathbb{R}^2$ into $h_m \times w_m = n_m \in \mathbb{N}$ tiles, and associating each tile $j \in 1, \ldots, n$ with a position $\mathbf{p}^{(j)} \in \mathbb{R}^2$ and an occupancy value $m^{(j)} \in \mathbb{R}$ ($m^{(j)} = 1$ if occupied, and $m^{(j)} = -1$ if free). While Gaussian density estimate obtained from the KF provides a continuous real value, by applying a threshold function which returns 1 for positive input and -1 for negative input, the map state can be estimated as a binary value. Using the index set notation of (2), the set of map tile indices within the field of view can be described as $\mathcal{I}_{\mathcal{F}}(\mathbf{x}, \{\mathbf{p}^{(j)}\})$. Using the image sensor from the flying robot, the occupancy within the FoV can be measured directly. The sensor state is given by $\mathbf{z}_k^{\text{map}} = [\{\mathbf{z}^{\text{map}}(m^{(j)})\}_{j \in \mathcal{I}_{\mathcal{F}}(\mathbf{x}_k, \{\mathbf{p}^{(j)}\})}]$, and the sensor model is set as $\mathbf{z}^{\text{map}}(m^{(j)}) = m^{(j)} + v$, where $v \sim \mathcal{N}(0, \sigma)$. Similarly to the landmark localization problem above, we consider a unified exploration and landmark localization problem by computing a control policy π, mapping a state $\mathbf{x}_k$ and observations $\mathbf{z}_{0:k} = [\mathbf{z}_{0:k}^{\text{land}\top}, \mathbf{z}_{0:k}^{\text{map}\top}]^\top$ to an input $\mathbf{u}_k$, that solves:

$$\max_{\pi} \mathbb{E}_{\pi} \left(\sum_{k=0}^{\infty} \gamma^k \left(\rho \alpha^{\text{land}} r_k^{\text{land}} + (1 - \rho) \alpha^{\text{map}} r_k^{\text{map}} \right) \right), \quad (13)$$

with $r_k^{\text{map}} = \log\det(\bar{Y}_{k+1}) - \log\det(\bar{Y}_k)$, $\bar{Y}_{k+1}^{(j)} = \bar{Y}_k^{(j)} + \frac{1}{\sigma^2} \boldsymbol{\xi}_k^{(j)}$, $\boldsymbol{\xi}_k^{(j)} = 1 - \Phi(d(\mathbf{q}(\mathbf{x}_k, \mathbf{p}^{(j)}), \mathcal{F}))$, where $\bar{Y}_k^{(j)} \in \mathbb{R}_+$ is a j-th diagonal element in the information matrix $\bar{Y}_k$ of the volumetric map, subject to the models in (1)–(3). The hyper parameter $\rho \in [0, 1]$ controls the weight of each reward, and $\alpha^{\text{land}}, \alpha^{\text{map}}$ are normalization factors. One notable difference from the landmark scenario is that, the set of indices $\mathcal{I}_{\mathcal{F}}(\mathbf{x}, \{\mathbf{p}^{(j)}\})$ is independent from the map state $\mathbf{m}$, thereby the posterior mean of the Kalman Filter is not needed in the observation space.

The state vector $\mathbf{s}_k \in \mathbb{R}^{n_x + 2n_l + n_\lambda + 2n_m}$ consists of $\mathbf{s}_k^{\text{land}} \in \mathbb{R}^{n_x + 2n_l + n_\lambda}$ of the landmark localization task (11) and $\mathbf{s}_k^{\text{map}} \in \mathbb{R}^{2n_m}$ of the map exploration. The state $\mathbf{s}_k^{\text{map}}$ includes the agent's differentiable FoV $\boldsymbol{\xi}_k^{(j)}$ and the current information of each pixel in the map $\bar{Y}_k^{(j)}$: $\mathbf{s}_k = \left[\mathbf{s}_k^{\text{land}\top}; \mathbf{s}_k^{\text{map}\top}\right]^\top, \mathbf{s}_k^{\text{map}} = \left[\{\boldsymbol{\xi}_k^{(j)}\}_{j=1}^{n_m\top} \{\bar{Y}_k^{(j)}\}_{j=1}^{n_m\top}\right]^\top$. The two states for landmark localization and volumetric exploration are processed independently to extract their corresponding feature vectors: $\mathbf{s}_k^{\text{land}}$ goes through the same attention-based network without the last fully-connected layer, while $\mathbf{s}_k^{\text{map}}$ is reshaped as a 2-channel image $S_k^{\text{map}} \in \mathbb{R}^{2 \times h_m \times w_m}$ and compressed by two convolutional layers followed by a flatten operation. The two feature vectors are concatenated and fed to two fully-connected layers to compute the action or value.

IV. Evaluation

We consider aerial surveillance of a 3-D environment using a flying robot with downward-facing RGB-D sensor. For simplicity, we assume that the orientation and height of the robot are not controlled, and the $x - y$ position of the robot can be controlled directly by linear velocities, while the landmarks are located in a 2-D plane with the same height. Hence, the robot state is the position $\mathbf{x}_k \in \mathbb{R}^2$ and the control input is the linear velocity $\mathbf{u} \in \mathbb{R}^2$ so that $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k$. The robot's FoV is a circle with a fixed radius on the ground. For sensing of landmarks, we suppose both range and bearing sensors are available, thereby the sensor model is described by the robot-body-frame coordinates, i.e, $H = I_2$ and $V = \sigma^2 I_2$ with a sensor noise magnitude $\sigma \in \mathbb{R}_{\geq 0}$. Under this setting, it is easily shown that the information matrix Y_k by Kalman Filter (5) and differentiable FoV (9) becomes a diagonal matrix, and thus $\boldsymbol{\lambda}_k = \text{diag}(Y_k) \in \mathbb{R}^{2n_l}$ is sufficient for the information vector in observation space. At each episode, the initial estimations of landmark positions $\boldsymbol{\mu}_0$ are sampled from a Gaussian distribution, while its mean value corresponds to landmarks' true positions, and the variance is determined by the sensor noise.

We demonstrate the performance of our approach in several simulations. First, we show qualitative and quantitative results for landmark localization. With non-uniform initial information values among the landmarks, the agent prioritizes landmarks with low information. Then, promising results for the joint exploration and landmark localization method are presented. Finally, we apply our method in a high-fidelity Unity simulation showing its potential for transfer in a real-world setting.

A. Evaluation Settings

We first conduct landmark localization experiments with 3, 5, and 8 randomly scattered landmarks in the environment. Respectively, the agents takes 8, 15, and 18 time steps per episode in each scenario during the training, so that the agent requires an efficient exploration by the terminal time. At the beginning of each episode, the agent's position is chosen randomly using a uniform distribution, $\mathbf{x}_0 \sim \text{Uniform}([-2, 2] \times [-2, 2])$, while the landmark positions are specified using a uniform distribution within $[-8, 8] \times [-8, 8]$, $[-10, 10] \times [-10, 10]$ and $[-12, 12] \times [-12, 12]$, respectively. The control scaling factors are set as 3, namely, the neural network stochastic control policy $\mathbf{u}_k \sim \boldsymbol{\pi}_\theta(\cdot|\mathbf{s}_k)$ is updated within the range $\mathbf{u}_k \in [-3, 3] \times [-3, 3]$. Regarding the hyper-parameters in Sec. III-B, we chose a smoothing factor κ, sensing radius, and sensor noise σ as 0.5, 2 and 0.5, respectively. The models are trained for a million time steps using a 24G NVIDIA GeForce RTX 3090 GPU. Specifically, the training process for each model was completed within 30 minutes in our experiment.

B. Landmark Localization Quantitative Comparisons

One demonstration of the trajectories generated from our learning policy in three different maps is depicted in Fig. 3, which illustrates that the agent traverse through all the landmarks. We compare our method, named *PPO-att*, to two baselines: (1) *PPO-mlp*: a policy network replacing the

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.

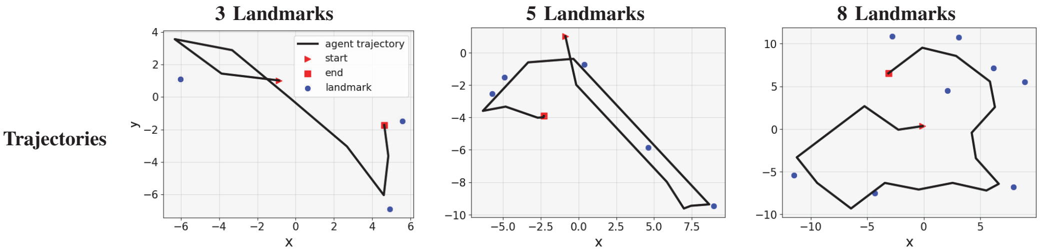

Fig. 3: Landmark localization trajectories. At test time, the agent is randomly initialized near the origin and has to explore the map to reduce uncertainty in the landmark positions with a limited sensing radius of 2. The blue dots indicate the real positions of the landmarks. The small red triangle and square show the agent's initial and current positions, respectively.

attention block with a multi-layer perceptron and (2) *iCR-landmark*: the open-loop optimal control method iCR [14], which performs finite-horizon trajectory optimization. We use the same hyper-parameters, protocols, and landmark and agent initial positions for all methods.

Each method is tested for 30 episodes per environment. Note that *PPO-att* and *PPO-mlp* are both trained with three random seeds to get three models in each environment. Each model runs 10 episodes in each map. We utilize two metrics for quantitative comparisons: (1) *Reward*: mean and standard deviation values of the cumulative reward, and (2) *MAE*: the mean absolute error of the landmark localization at the end of each episode. Both metrics are computed over 30 episodes. We also tested the methods in environments with and without additive Gaussian noise in the robot dynamics (1). The results are shown in Table I.

iCR-landmark is capable of finding an optimal control sequence after sufficient number of iterations. However, it is not able to generalize the control input to different ground truth of the landmark positions, and its performance is highly dependent on a precomputed initial control sequence. The iCR-landmark trajectory can be recomputed at run-time, but such approach is out of our scope aiming to generalize a precomputed policy to new environments. Secondly, according to the acquired data, *PPO-mlp* performs relatively well when there are only three landmarks but obtains the worst results when there are more landmarks. Intuitively, although *PPO-mlp* is able to handle landmark randomization, it cannot extract relationship features between landmarks and the agent. Thus, when the amount of the landmarks increases, exploration fails and leads to a significant performance drop. In contrast, our method always performs the best among other compared methods and is able to localize randomized landmarks with a small mean absolute error at the end of each episode. It is noteworthy that *PPO-mlp* also exhibits an advantage over other methods in scenarios with agent motion noise. They demonstrated the robustness of the proposed method in noisy dynamical environments.

C. Landmark Localization with Non-uniform Information

We conduct a supplementary experiment to further demonstrate the features and advantages of our method by setting non-uniform initial information, where one landmark has much higher information (named *high-info landmark* hereafter) than others. Fig. 4 shows different trajectories generated by the learned control policy for different landmark configurations. We can clearly see that the agent always prioritizes its exploration to landmarks with lower initial information values, and even ignores the *high-info landmark* due to the limited number of time steps. With these experiments, we verify that our approach is capable of prioritizing less certain landmarks.

TABLE I: Quantitative Comparison Results. The table shows the average and standard deviation for the cumulative episode rewards (higher is better) and the average estimation error after mapping (lower is better). Tests are performed in three kinds of maps with or without motion noise.

Method		3 Landmarks		5 Landmarks		8 Landmarks	
		Reward	MAE	Reward	MAE	Reward	MAE
iCR-landmark	w/o noise	11.93 ± 6.71	0.37	20.57 ± 7.58	0.34	29.18 ± 10.0	0.36
	w/ noise	13.3 ± 5.32	0.37	18.43 ± 6.48	0.39	25.31 ± 7.69	**0.34**
PPO-mlp	w/o noise	16.39 ± 4.82	0.34	18.97 ± 6.23	0.36	16.36 ± 10.39	0.37
	w/ noise	16.27 ± 4.9	**0.32**	15.26 ± 6.39	0.36	16.78 ± 9.64	0.37
PPO-att	w/o noise	**18.54** ± **2.9**	**0.29**	**30.27** ± **3.0**	**0.31**	**38.25** ± **8.0**	**0.35**
	w/ noise	**18.13** ± **3.3**	**0.32**	**26.53** ± **6.17**	**0.33**	**30.15** ± **11.03**	0.36

D. Joint Exploration and Landmark Localization

We also evaluate the joint exploration and landmark localization method described in Sec. III-D. The environment size is set as 30×30. During training, the agent is randomly initialized with a uniform distribution on $[13, 17]$ for both the x and y axes. The initial positions of the 5 landmarks are also uniformly randomized in a larger x, y range of $[5, 25]$. The time horizon is fixed at 15 steps per episode. The rest of the environment and training parameters are the same in Sec. IV-A.

We compare two policies trained with different reward weights ρ in (13). The weight of the *exploration-exploitation policy* is set to $\rho = 0.2$, i.e., the agent is pursuing both map exploration and landmark localization exploitation. The weight in the *exploitation policy* is set to $\rho = 1.0$, i.e., the agent is only exploiting the landmark localization since only r_k^{land} is kept in the reward.

To demonstrate the agent's ability of localizing the landmarks and exploring the map simultaneously, the 5 landmarks are randomly initialized in a more concentrated area (e.g. $[20, 25]$ for x and y in Fig. 5). Exploration-exploitation policy is expected to visit the landmarks first to obtain high information initially, and then explore the rest of the map to continue gaining information. In contrast, the exploitation policy is expected to remain around the landmarks leaving the majority of the map unexplored.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.

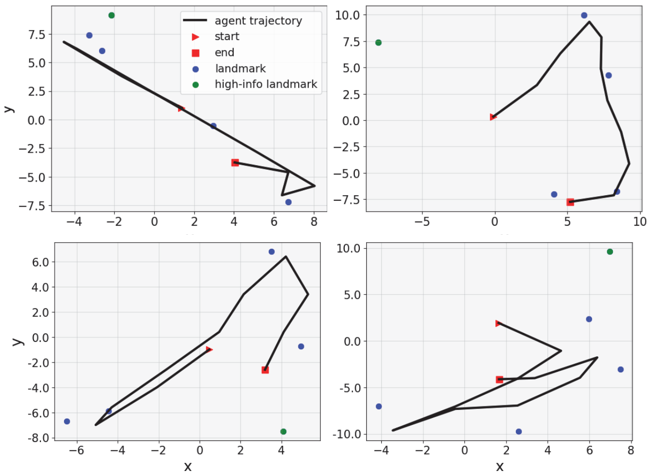

Fig. 4: Trajectories under non-uniform initial information values. When there is one landmark with higher initial information, the agent prioritizes the sensing of other landmarks with lower information. The green dots illustrate landmarks with higher information.

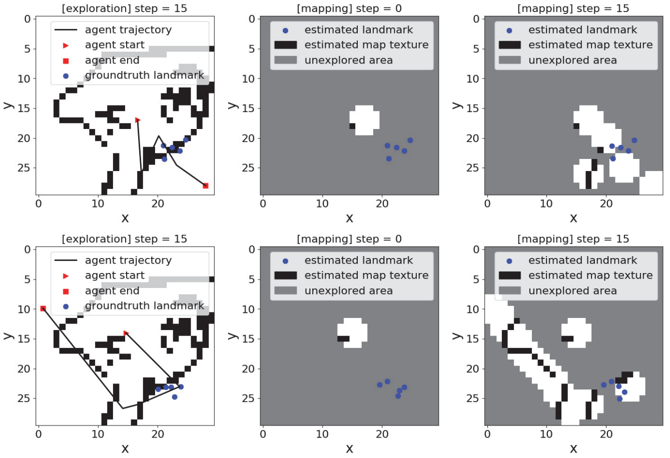

Fig. 5: Landmark localization and exploration using an exploitation policy (top) and an exploration-exploitation policy (bottom). The left column shows the final trajectory on the ground-truth map. The two columns on the right show the estimated landmark positions and occupancy map at the beginning and the end of the episode.

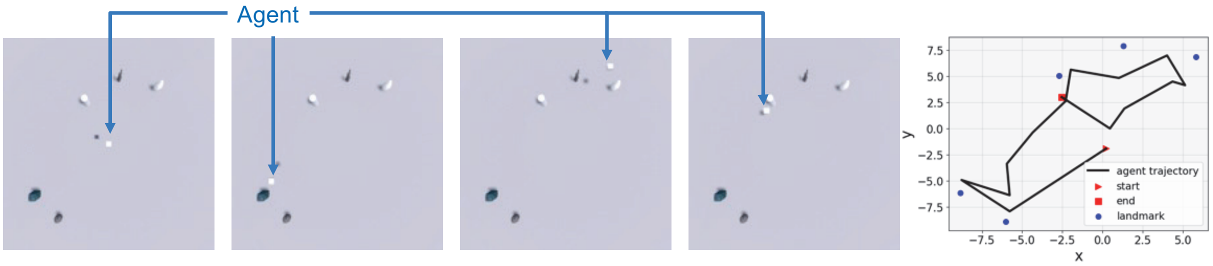

Fig. 6: Snapshots at time steps 0, 5, 10 and 15 in an active landmark localization episode in the Unity simulator.

Fig. 5 shows the test results for the exploitation policy (top) and the exploration-exploitation Policy (bottom). It is obvious that the agent employing the exploration-exploitation policy continues to explore the map after detecting the landmarks, while with the exploitation policy the agent stays only around the landmarks. Qualitatively, the landmark positions are better estimated by the exploitation policy at the last step because it focuses on maximizing the information gain of only the landmarks.

E. Unity Simulation

To examine the applicability of the method in real-world environments, we designed a Unity simulation [7] which provides realistic training and testing settings. We used 5 landmarks represented by different objects in Unity and positioned randomly in a bounded area in each episode. An aerial agent explores the environment to localize the scattered landmarks, as illustrated in Fig. 1. The agent is equipped with a downward-facing aligned semantic and depth sensor with Gaussian noise on the depth values. When a landmark object is inside the sensor FoV, the agent uses the semantic map to extract the pixels of interest. Then, based on the depth values of the extracted pixels and the intrinsic and extrinsic camera matrices, we obtain a 3-D point cloud observation of object and estimate its center position in the plane. To guarantee the same shape of the sensor model, we also apply a squared signed distance function for the calculation. Fig. 6 illustrates the agent exploring the environment. *PPO-att* trained in a 2-D environment achieved a desired trajectory visiting all the landmarks.

V. Conclusion

This paper proposed a learning method for active landmark localization and exploration with an information-theoretic cost over continuous control space. Key aspects of our method include (i) reward shaping using a differentiable FoV, (ii) attention-based neural network architecture for landmark prioritization, and (iii) joint landmark localization and environment exploration. Future work will focus on deterministic computation of reward gradients, collision and occlusion modeling, and deployment on a real robot system.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *arXiv preprint arXiv:2207.00254*, 2022.

[3] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive computing*, vol. 3, no. 4, pp. 24–33, 2004.

[4] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.

[5] K. D. Julian and M. J. Kochenderfer, "Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1768–1778, 2019.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[7] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, *et al.*, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[8] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.

[9] A. Elfes, "Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework," in *International Workshop on Reasoning with Uncertainty in Robotics*. Springer, 1995, pp. 91–130.

[10] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "Active SLAM and exploration with particle filters using Kullback-Leibler divergence," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, 2014.

[11] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4775–4782.

[12] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using cauchy-schwarz quadratic mutual information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4791–4798.

[13] Z. Zhang, T. Henderson, S. Karaman, and V. Sze, "FSMI: Fast computation of Shannon mutual information for information-theoretic mapping," *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1155–1177, 2020.

[14] S. Koga, A. Asgharivaskasi, and N. Atanasov, "Active exploration and mapping via iterative covariance regulation over continuous *SE(3)* trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2735–2741.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[18] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 31–36.

[19] K. Lobos-Tsunekawa, F. Leiva, and J. Ruiz-del Solar, "Visual navigation for biped humanoid robots using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3247–3254, 2018.

[20] H. Jeong, B. Schlotfeldt, H. Hassani, M. Morari, D. D. Lee, and G. J. Pappas, "Learning q-network for active information acquisition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6822–6827.

[21] C. D. Hsu, H. Jeong, G. J. Pappas, and P. Chaudhari, "Scalable reinforcement learning policies for multi-agent control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4785–4791.

[22] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," *arXiv preprint arXiv:1903.01959*, 2019.

[23] F. Chen, J. D. Martin, Y. Huang, J. Wang, and B. Englot, "Autonomous exploration under uncertainty via deep reinforcement learning on graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6140–6147.

[24] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[25] M. Lodel, B. Brito, A. Serra-Gómez, L. Ferranti, R. Babuška, and J. Alonso-Mora, "Where to look next: Learning viewpoint recommendations for informative trajectory planning," *arXiv preprint arXiv:2203.02381*, 2022.

[26] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[27] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[28] Q. Long, Z. Zhou, A. Gupta, F. Fang, Y. Wu, and X. Wang, "Evolutionary population curriculum for scaling multi-agent reinforcement learning," *arXiv preprint arXiv:2003.10423*, 2020.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on July 14,2023 at 20:57:51 UTC from IEEE Xplore. Restrictions apply.