Policy Learning for Visually Conditioned Tactile Manipulation

Tarık Keleştemur[1], Taşkın Padır[1] and Robert Platt[2]

***Abstract*— Recent work on robot learning with visual observations has shown great success in solving many manipulation tasks. While visual observations contain rich information about the environment and the robot, they can be unreliable in the presence of visual noise or occlusions. In these cases, we can leverage tactile observations generated by the interaction between the robot and the environment. In this paper, we propose a framework for learning manipulation policies that fuse visual and tactile feedback. The control problems considered in this work are to localize a gripper with respect to the environment image and navigate to desired states. Our method uses a learned Bayes filter to estimate the state of a gripper by conditioning the tactile observations on the environment image. We use deep reinforcement learning for solving the localization and navigation problems provided with the belief of the gripper's state and the environment image. We compare our method against two baselines where the agent uses tactile observation directly with a recurrent neural network or uses a point estimate of the state instead of the full belief state. We also transfer the policies to the real world and validate them on a physical robot.**

I. INTRODUCTION

It is not yet clear how best to leverage force and tactile measurements in order to improve the performance and reliability of robotic manipulation. Compared with camera images, tactile and force data contain much less information per measurement. However, some types of information like whether the robot is contacting the environment and how much force is applied is difficult to estimate based on visual information alone. Ideally, we would like to combine visual and tactile/force information, leveraging the strengths of each. However, it is not clear how to accomplish this. Recently, we proposed an approach that uses sequential Bayes filtering to estimate robot position based on tactile/force data relative to an reference visual image of the scene [1]. Here, it is assumed that a reference image is obtained prior to tactile interaction that shows the overall scene. As force/tactile information is perceived, a conditional Bayes filter estimates robot pose relative to the reference image. This approach is particularly useful in unstructured domains containing novel objects. If we were to attempt to estimate the pose and shape of all potentially relevant objects in a scene using tactile/force data alone, this would require us to track a probability distribution over a very high dimensional space. The visually conditioned localization method described above avoids the high dimensional estimation problem by partitioning the state space into world state that is represented by the reference image and robot state that is estimated using filtering.

This paper addresses a key question in this setting – given initial uncertainty about the position of a robot in a scene, can we learn a policy that enables the robot to localize itself or to reach a desired position as quickly as possible? In contrast to prior work that studies this problem in the unconditioned setting (i.e. studying this problem as a POMDP), we learn policies that are conditioned on visual input. Compared to a setting where the policy is trained for in a specific manipulation setting, this is a harder learning problem because our agent must learn to localize or navigate in an arbitrary environment. One perspective on this is that we are casting robotic manipulation as a mixed observability Markov decision process (a MOMDP [2]) rather than a POMDP, as in e.g. [3]. The state of the world is assumed to be fully observed (in the form of the reference image) and the robot position is partially observed via the tactile/force measurements.

We approach the problem by applying standard deep reinforcement learning (RL) methods to the MOMDP problem. The state of the RL agent is taken to be the combination of the reference image and the belief state of robot pose with respect to that image as estimated by the visually conditioned sequential Bayes filter. We evaluate the approach in a setting where a robot manipulator must localize itself through force interactions with objects and navigate to desired goals to solve manipulation tasks. To this end, we designed three experiments: an active localization task, a navigation task where the objective is to reach to a desired state, and a drawer opening task where the aim is to place the gripper on drawer handle and open it under visual and kinematic noise. We compare our belief state approach with two alternative approaches in simulation: a baseline that replaces the full belief state with a point estimate, and a baseline that replaces the belief state with recurrent neural network (GRU) components. In both cases, we find that the belief based approach significantly outperforms on all the tasks. Finally, we demonstrate that the simulated results translate well to a real robotic setting. The code and videos are available at https://sites.google.com/view/vctm/.

II. RELATED WORK

A. Localization with Touch

The early work of [4], [5], [6], [7], [8], [9] are the first examples of object pose estimation using Bayes filters. These methods aim to find the location of the object using fingertip contact sensing with fixed observation and transition

*This material is based upon work supported by the National Science Foundation under Award No. 1544895, 1928654, 1935337, 1944453, 1952032.

[1]Tarık Keleştemur, and [1]Taşkın Padır are with the College of Engineering, and [2]Robert Platt is with the Khoury College of Computer Sciences, Northeastern University, Boston, Massauchusetts 02115, USA. kelestemur.t@northeastern.edu

functions. Liang et al. [10] focus on a similar problem but also consider the motion dynamics of the object while manipulating in hand. Pfanne et al. [11] combines tactile feedback with visual feedback to improve the object localization accuracy. Similarly, [12] fuses visual and tactile sensing to estimate not only object poses but also the end-effector pose. In this setting, the object and the gripper must be in the scene of the camera which might not be always possible due to occlusions by the arm or the gripper. It is important to rely only on tactile feedback during the object-robot interactions to avoid occlusions. Although these methods work well in practice, they are not generalized to novel objects due to their reliance on prior knowledge of the objects. With the recent progress in the high-resolution tactile sensors (e.g. GelSight [13] or OmniTact [14]), a series of work focus on visual-tactile localization of objects. Li et al. [15] use the GelSight sensor to localize the pose of a USB stick by using the RANSAC algorithm. Izatt et al. [16] combines the GelSight sensor information with point clouds and performs the Iterative Closest Point (ICP) method for tracking the pose of the object. The objective of all the work described above is to localize the pose of an object with respect to the gripper. The dual of this problem is to localize the gripper with respect to the environment. Platt et al. [17] localizes the pose of a robotic hand with respect to a flexible piece of plastic textured. Similarly, [18] propose to use a visual-tactile sensor to match the features of the sensor reading with the pre-generated features of a fixed environment image to localize the gripper. The feature matching is done by the scale-invariant feature transform (SIFT) method [19]. In both of these works, the environment is fixed and known beforehand and the localization cannot generalize to new objects or environments. On the other hand, our method is able to generalize to a variety of scenes and unseen objects. We achieve this by conditioning the belief updates on the image of the environment taken prior to the interactions.

B. Differentiable Bayes Filters

In a partially observable environment, the agent does not have access to the full state information, instead, it receives a type of observation from the environment that relates to the underlying state. For example, a robot might use images of objects to grasp them in place of their exact geometries and poses. Bayes filters are a family of well-established algorithms that can estimate the agent's state from observations and track it over time [20]. A Bayes filter works by maintaining a belief of the state and recursively update the belief using the observation and transition functions. Recent line work has shown that one can learn the observation and transition functions if they are not known. Jonschkowski and Brock [21] proposed an end-to-end learnable Bayes filter that represents the belief with a histogram. Jonschkowski et al. [22] and Karkus et al. [23] concurrently introduced differentiable particle filters for continuous states. A learnable Kalman filter with Gaussian belief is presented by Haarnoja et al. [24]. Similarly, Karl et al. [25] and Watter et al. [26] proposed to learn latent spaces with Gaussian beliefs. All the methods mentioned above report their results in simulated environments with a focus on agent localization. In contrast, we focus on touch localization and reaching tasks, furthermore, our models are trained in simulation and transferred to the real world.

C. RL under Partial Observability

Deep reinforcement learning has shown great success for solving many robotic tasks in the last few years, however, policy learning under partial observability is still an open challenge. There are two common approaches to handling this problem in the literature. The first is to use recurrent neural networks for learning policies over histories over observations and actions. Hausknecht et al. [27] introduced Deep Recurrent Q-Network (DRQN) which is an extension of the DQN algorithm [28] where the Q-function is constructed using Long Short Term Memory (LSTM) layers [29]. This idea is later applied to on-policy methods [30] as well as model-based deep RL [31], [32]. The second approach is to use beliefs where the RL agent would get the current belief at every time step. Karkus et al. [33] combines the QMDP planner (an approximate POMDP solver [34]) with a differentiable Histogram filter to solve several partially observable tasks and show strong results compared to recurrent networks. Chaplot et al. [35] and Gottipati et al. [36] took a similar approach where they learned policies with belief inputs for solving active localization problem for mobile robots. Recently, a work by Wirnshofer et al. [37] showed promising results for contact-rich tasks under partial observability. Their method uses a particle filter to track the positions and velocities of the system and train a DQN agent with belief as input. In our work, we follow the second approach because as it has shown in the literature, using an algorithmic prior such as Bayes filters can provide faster learning and increase the performance when compared to recurrent networks. In our experiments, we validate this hypothesis as well. While these methods mostly focus on learning policies for visual localization in simulated environments, we focus on the problems of localizing a robotic gripper and reaching to desired goals with touch feedback.

III. BACKGROUND

Partial Observability: If the agent does not have access to the complete state information, the problem can be formulated as a partially observable Markov Decision Process (POMDP) [38]. A POMDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O})$ where $\mathcal{S}, \mathcal{A}$, and Ω are the state, action, and observation spaces, respectively. The agent acts in the environment by taking an action $a_{t-1} \in \mathcal{A}$ and move from the previous state $s_{t-1} \in \mathcal{S}$ to the next state $s_t \in \mathcal{S}$ by following the state-transition function $\mathcal{T}(s_t, a_{t-1}, s_{t-1}) = p(s_t|s_{t-1}, a_{t-1})$. After each transition, the agent receives an observation $o_t \in \Omega$ and a reward $r_t \in \mathbb{R}$ provided by the reward function $\mathcal{R}(s_t)$. The observation function $\mathcal{O}(o_t, s_t, a_{t-1}) = p(o_t|s_t, a_{t-1})$ defines the probability of receiving the observation o_t after taking action a_{t-1} and landing in state s_t.

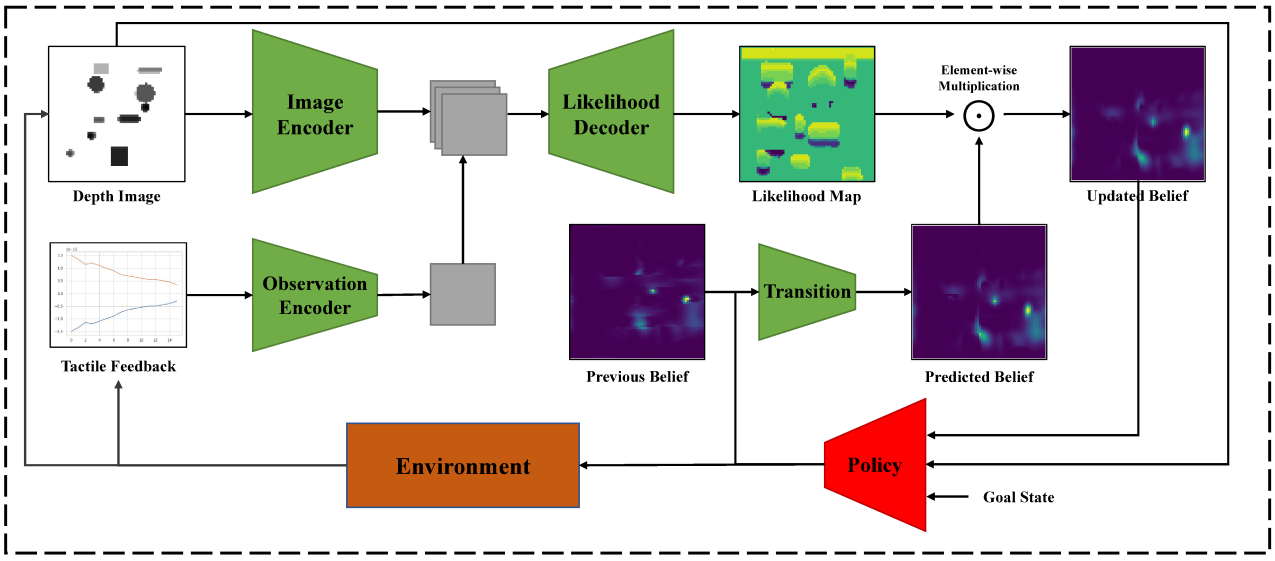


Fig. 1. The depth image and the tactile feedback are fed into their encoders. The output of these encoders are concatenated and passed through the likelihood decoder which generates the observation probability map. The transition layer predicts the belief at next time step which is then multiplied with the observation probability to produce the updated belief. Finally, the belief, the depth image and the goal state are fed into the policy.

Bayes Filters: In order to address the partial observability, an agent can estimate a probability distribution over the state space (also called *belief*) using recursive Bayes filters [20]. A Bayes filter estimates the belief over states by conditioning on the past observations and actions: $bel(s_t) = p(s_t|a_{1:t-1}, o_{1:t})$. The belief is updated at each time step by taking the *prediction* step and *observation update* step:

$$bel(s_t) = \underbrace{\eta \mathcal{O}(o_t, s_t, a_{t-1})}_{\textit{Observation Update}} \underbrace{\sum_{s_{t-1} \in S} \mathcal{T}(s_t, a_{t-1}, s_{t-1}) bel(s_{t-1})}_{\textit{Prediction Update}}$$

where η is the normalization factor.

Goal Conditioned RL: In goal-conditioned RL, each episode starts with sampling an initial state $s_0 \sim p_0$ and goal state $g \sim p_g$ where p_0 is the initial state distribution and p_g is the goal distribution [34]. The goal stays fixed throughout the episode. The objective is to find a goal-conditioned policy $\pi(a_t|s_t, g)$ that maximize the expected discounted return: $\mathbb{E}_\pi[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}(s_t, g)]$ where T is the maximum horizon and , $\gamma \in [0, 1]$ is the discount factor, and $\mathcal{R}(s_t, g)$ is the goal-conditioned reward function.

IV. PROBLEM STATEMENT

A. Setting of the Problems

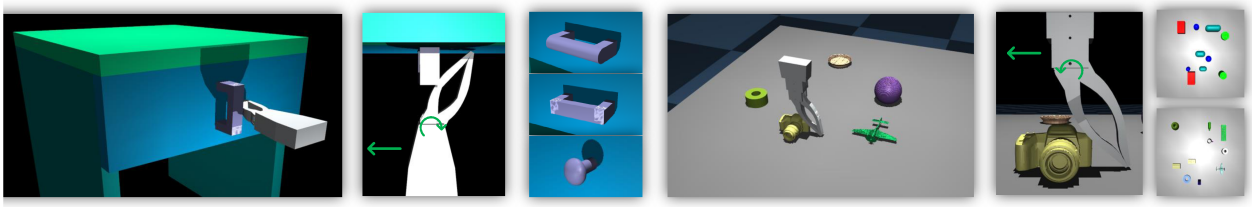
We assume we are given an environment that includes a set of objects with unknown geometry and locations, a robotic hand that moves in a parallel plane, and a depth camera looking towards the objects. As the hand moves in a plane, the fingers make contact with the objects, thereby displacing the fingers and producing force or tactile measurements. A simulation of this scenario is shown in Fig. 2. Our objective will be either to localize the hand or to reach a desired position with respect to the depth image. The position of the hand in the plane above the table is a point in $\mathbb{R}^2$. The depth image is an image I with size $H \times W$. The tactile/force observation is $o \in \Omega$. In our setting, o is a short sequence of τ hand joint velocity measurements in $\mathcal{O} = \mathbb{R}^\tau$ produced by an actively compliant hand [39], however, our framework allows o to be virtually any sort of force/tactile measurement. We model this system as a discrete-time mixed observability Markov decision process (i.e. a MOMDP) [2]. In the MOMDP, some elements of state are fully observed on each time step while others are partially observed. In our case, we consider the depth image I to be fully observed and the hand position in the plane to be partially observed. To simplify things slightly, we will model hand position as $s \in \mathbb{Z}^{|H| \times |W|}$, the position of the corresponding pixel in I.

B. Problems

There are two problems associated with the MOMDP that we are interested in solving. The first is the inference problem (which is explored in our previous work [1]) where we infer hand position $p(s_t|h_{1:t}, I)$ given the depth image I and $h_{1:t} = (o_t, a_{t-1}, \ldots, a_1, o_1)$, a history of observations and actions. The second is a control problem where we must find a policy $\pi(a_t|h_{1:t}, I)$ that optimizes a reward function. Below, we explore the control problems in the force/tactile setting. We say that these problems are "visually conditioned" because we condition on the depth image I.

Definition 1: Visually Conditioned Active Tactile Localization. Given a depth image I of a novel scene, find a policy $\pi(a_t|h_{1:t}, I)$ that localizes the gripper in a minimum number of time steps.

Definition 2: Visually Conditioned Tactile Navigation. Given a depth image I of a novel scene, find a policy $\pi(a_t|h_{1:t}, I)$ that reaches a goal position in a minimum number of time steps.



(a) Drawer Opening Environment

(b) Localization and Navigation Environment

Fig. 2. The simulation environments for the localization, navigation and drawer opening tasks. The trained agents can open a drawer with different types of handles (a), and localize or navigate to a desired position on a tabletop with novel objects (b).

V. METHODS

We solve the active localization and navigation problems by formulating the MOMDP as a belief MDP, where the belief state is a probability distribution over the underlying state, $b_t = p(s_t|h_{1:t}, I)$.

A. Visually Conditioned Tactile Localization

In order to formulate the belief MDP, we first need to be able to track belief state, $b_t = p(s_t|h_{1:t}, I)$. We accomplish this by following the method developed in our prior work [1] where we train visually conditioned process and observation models independently of each other and then combine them using sequential Bayes filtering. We represent the observation and the transition functions as layers of neural networks and train them from data generated in simulation. Let $f_{\mathcal{O}}(\cdot)$ be a neural network that takes the environment image, tactile observation and the action as input and generates the likelihood probabilities of the current observation, i.e. $f_{\mathcal{O}}(o_t, a_t, I) = p(o_t|s_t, a_t, I)$. Let $f_{\mathcal{T}}(\cdot)$ be a neural network that takes the previous belief and the action as input and predicts the belief at the next timestep, i.e. $f_{\mathcal{T}}(bel(s_t-1), a_{t-1}) = \overline{bel}(s_t)$. The state space is defined as the projected pixel coordinates of the gripper in the environment image: $s_t = (p_x, p_y) \in \mathcal{Z}^{H\times W}$ where H is the height and W is the width of the image, I. The belief is encoded as a $H \times W$ matrix and computed as:

$$bel(s_t) = \eta f_O(o_t, a_t, I) \odot f_M(bel(s_{t-1}), a_t)$$

where $\odot$ is element-wise multiplication. Note that the observation function is conditioned on the environment image. This way, our method can generalize over novel configurations (shapes, sizes, and positions) of objects. The neural network that represents the observation function $f_{\mathcal{O}}$ is based on the U-net architecture [40] and consists of 3 modules: the image encoder, the observation encoder and the likelihood decoder. The image of the environment and the tactile feedback are fed into their encoders. The outputs of these encoders are concatenated and fed into the likelihood decoder to generate likelihood maps. For the transition function, we use a single 2-D convolutional layer which predicts the next belief. The predicted belief and the likelihood map are then multiplied element-wise and normalized to produce the belief at the next time step. The flowchart of the framework can be seen in Fig 1.

B. Policy Learning

We choose Proximal Policy Optimization (PPO) algorithm [41] for learning our policies due to its robustness to hyper-parameters and sample efficiency. The PPO simultaneously learns a stochastic policy and a value function approximation. To avoid large policy updates, which can cause performance drops, the PPO method limits the policy changes by clipping the objective function. We use the network architecture from [30] both for the policy and the value function. The weights of the policy and the value function are shared and optimized together.

Environment: We developed a simulation environment using the MuJoCo physics engine [42]. We use a gripper with hydro-static linear actuators [39] which allows us to set the finger joint stiffness to low values. This way, the gripper can interact with the objects without moving them. A depth camera is positioned towards the objects and captures the environment image prior to tactile interaction. In order too find the pixel coordinates of the gripper, we first transform the pose of the gripper base to the camera frame and then project it into pixel coordinates: $p = M_{int}M_{ext}P_w$ where M_{int} and M_{ext} are intrinsic and extrinsic camera matrices, respectively, $p = (p_x, p_y)$ is the pixel coordinates of the gripper, and P_w is the 3D position of the gripper's base in the environment.

Tasks: We evaluate our method on three manipulation tasks:

Active Localization: In this task, the gripper is positioned over a table and moves in a plane parallel to the table. (see Fig. 2) The objective is to localize the gripper in a minimum number of steps. The input the PPO agent is a 2-dimensional image where the first channel is the depth image of the environment taken prior to the interaction and the second channel is belief at current time step: $\pi(a_t|bel(s_t), I)$. Since the agent has no prior information about the location of gripper, the initial belief is uniform distribution over the state space: $bel(s_0) = \frac{1}{|\mathcal{S}|}\forall s \in \mathcal{S}$. At each episode, we randomize the positions, shapes and sizes of the tabletop objects.

Navigation: The navigation task has the same environment setup except that the goal is now to reach a desired pixel. To that end, the agent receives an additional one-hot image representing the goal state: $\pi(a_t|bel(s_t), g, I)$. The goal state is sampled from a uniform distribution at the beginning of each episode. Similar to the active localization task, the belief is initialized uniform over the state space.

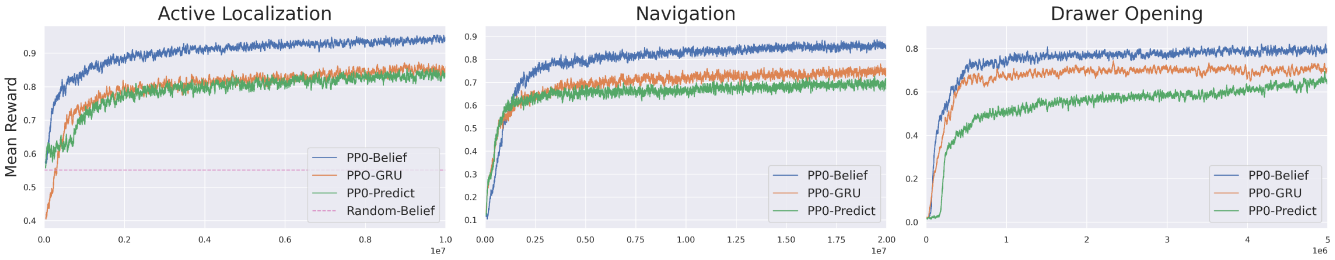


Fig. 3. Training performance for the Active Localization, Navigation and Drawer Opening tasks.

Drawer Opening: The goal of this task is to learn a policy that can open a drawer by placing the gripper on the handle. Similar to tasks above, the gripper moves in a plane parallel to the front surface of the drawer. Imagine a scenario where a mobile manipulator tries to open a drawer but due to noise from the visual detection of the handle and noise from the manipulator kinematics, the gripper is placed in a wrong position. In this case, the gripper can leverage tactile observations to reach the correct position of the handle. To realize this scenario, we first place the gripper in the correct position of the handle, then, we move the drawer by injecting noise up to 15cm in each direction. At each episode, we randomize the type of drawer handle and its sizes. The types of handles used in this task can be seen in Fig. 2. The initial belief is uniform over a set of 11×11 square states centered at the actual position of the gripper.

Actions: On each time step, the robot selects an action $a \in \{\text{NORTH}, \text{SOUTH}, \text{EAST}, \text{WEST}\}$ which moves the hand a short distance in the corresponding direction in the plane. In the case of drawer opening task, the agent has an additional action to command the gripper to grasp the handle and pull back to open the drawer. The gripper moves 5 pixels (approximately 5cm) in the corresponding direction with constant velocity. At each pixel movement, the belief is updated using the observation and transition function. The last belief is returned to the PPO agent.

Rewards: We use sparse reward functions where the agent gets a reward of 1 if the task is achieved and gets 0 otherwise. At any time during the task, the predicted state can be found as the point with the highest probability in the belief $\hat{s}_t = argmax(bel(s_t))$. For the active localization task, the reward function defined as: $r_t = 1$ if $|s_t - \hat{s}_t|_1 \leq \epsilon$ and for the navigation task the reward function defined as: $r_t = 1$ if $|s_t - g|_1 \leq \epsilon$ where g is the goal state, and ϵ is the error threshold. The drawer opening tasks gets a reward of 1 if the drawer is successfully opened. We simply check this condition by distance of the drawer before and after opening the drawer. The episode is terminated if one of the following conditions is met: the goal is reached, the gripper went out of the scene, or the maximum time limit is reached. The error threshold are selected as $\epsilon = 1$ for the active localization task and $\epsilon = 3$ for the navigation task. Note that 1 pixel is approximately 1cm in distance. The task horizon for the active localization and the navigation tasks is 32 steps whereas it is 16 steps for the drawer opening task.

VI. EXPERIMENTS

We first do a series of experiments in simulation to investigate our method's performance, generalization capabilities, and robustness to the noise of transition and observation functions. Later, we deploy our method on a real robot to show that the models trained in the simulation can be transferred to the real world without domain randomization or fine-tuning on real world data.

A. Simulation Experiments

We compare our method (which we call *PPO-Belief*) against three baselines. First is *PPO-GRU* in which the PPO agent uses a recurrent network. For this baseline, we use the same architecture with PPO-Belief for encoding the environment and goal images. The output features of this encoder are then concatenated with the tactile feedback and fed into an gated recurrent unit (GRU) [43] layer. In order to estimate the position of the gripper, we sill do the belief updates, however, the PPO-GRU agent does not have access to the belief and directly uses tactile observations. The second baseline is *Random-Belief* where the agent randomly selects actions at each time step and use belief to estimate the state. This baseline is only compared for the active localization task. The final baseline is *PPO-Predict* which is similar to PPO-Belief but instead of belief as the input, the agent takes a one-hot image of the predicted state. This baseline shows the importance of having the uncertainty as an input modality for solving these tasks. The Fig. 3 shows performance of the agents over course of training for all the tasks. As can be seen from the training curves, the PPO-Belief agent outperforms the baselines and converges to higher returns.

Generalization Experiments: To further investigate our method's generalization performance, we ran 10000 episodes for the active localization and navigation tasks under three different scenarios after training is completed. The scenarios are 1. an *uncluttered* scene where there are 4 primitive objects; 2. a *cluttered* scene where there are 10 primitive objects; 3. a *mesh* scene where there are 20 mesh objects from the 3DNet object dataset [44]. Note that we train our policies only on the uncluttered scene.

To show the generalization capabilities over object configurations, we randomly sample sizes and positions of objects at the beginning of each episode. For the 3DNet scene, 5 objects are randomly selected and placed on the table. We

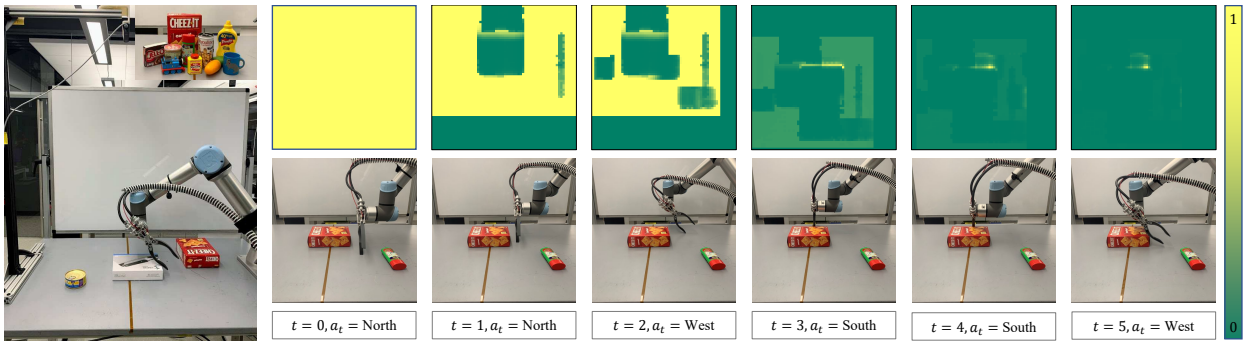


Fig. 4. Belief Sequence - As the gripper interacts with the objects (bottom row) the belief becomes less uncertain (top row) over time.

TABLE I

ACTIVE LOCALIZATION RESULTS OVER 10000 EPISODES

	Mean Episode Rewards		
	Uncluttered	Cluttered	Mesh
PPO-Belief	$\mathbf{95.13 \pm 0.21}$	$\mathbf{91.86 \pm 0.27}$	$\mathbf{86.02 \pm 0.34}$
PPO-GRU	85.65 ± 0.35	80.44 ± 0.39	76.20 ± 0.42
	Mean Episode Lengths		
	Uncluttered	Cluttered	Mesh
PPO-Belief	**8.46**	**8.27**	**12.06**
PPO-GRU	11.83	13.05	18.04

TABLE II

NAVIGATION RESULTS OVER 10000 EPISODES

	Mean Episode Rewards		
	Uncluttered	Cluttered	Mesh
PPO-Belief	$\mathbf{88.95 \pm 0.31}$	$\mathbf{89.59 \pm 0.30}$	$\mathbf{86.77 \pm 0.33}$
PPO-GRU	76.44 ± 0.42	75.23 ± 0.43	73.81 ± 0.28
	Mean Episode Lengths		
	Uncluttered	Cluttered	Mesh
PPO-Belief	**9.82**	**8.81**	**10.41**
PPO-GRU	12.33	12.29	16.40

report the mean success rate and the mean episode length in Table I and Table II for the active localization and navigation tasks, respectively. Our method outperforms both of the baselines in all scenarios. It can generalize to new positions and sizes of the primitive objects. Moreover, it still performs well for mesh objects with arbitrary shapes.

Noise Analysis: We also conduct a series of experiments to investigate how our method performs under observation and transition noise. For the transition function noise, the gripper would move in a direction other than commanded with a probability of n_t sampled from a uniform distribution $n_t \sim \mathcal{U}(0, 1)$. For the observation noise, we add a noise n_o to the tactile observations sampled from a Gaussian distribution $n_o \sim \mathcal{N}(\mu, \sigma^2)$ where we keep the mean μ as zero and vary the standard deviation σ. In Table III, we report the task success rate under different degrees of noise for the active localization and navigation tasks over 10000 episodes.

B. Real World Experiments

We transferred the policies trained for the active localization and navigation tasks in simulation to the real world for real robot experiments. The gripper is attached to a Universal Robot arm and it is moved with a Jacobian-based velocity controller. A Structure depth sensor is placed over the table. The depth image is pre-processed to filter out outlier pixels. We also applied a low-pass filter to the finger joint velocities to get rid of the noise. The experiment setup and the objects used in the real world experiments can be seen in Fig. 4. We use a total of 10 objects whose shape can be primitive or arbitrary. At the beginning of each episode, 2 objects are randomly selected and placed on the table. For both of the tasks, we ran 10 episodes and the error threshold is set to 3 pixels. For the active localization task, the gripper was able to localize 9 out of 10 runs. For the navigation task, the gripper was able to reach the goal for every run. In Fig 4, we show the belief and the gripper's position over time for a single run of the active localization task.

TABLE III

PERFORMANCE UNDER NOISE

Transition Noise (n_t)	Active Localization	Navigation
0.01	93.54 ± 0.24	85.57 ± 0.35
0.05	86.93 ± 0.33	76.69 ± 0.42
0.1	78.25 ± 0.41	64.06 ± 0.47
0.2	60.72 ± 0.48	43.49 ± 0.49
Observation Noise (σ)	**Active Localization**	**Navigation**
0.05	94.38 ± 0.23	86.78 ± 0.33
0.1	94.38 ± 0.23	86.46 ± 0.34
0.2	90.49 ± 0.29	81.64 ± 0.38
0.3	84.43 ± 0.36	74.89 ± 0.43

VII. CONCLUSIONS

We present methods for learning policies that can localize and navigate a robotic gripper to solve manipulation tasks under sensor noise. Our simulation results show that this approach outperforms recurrent-based methods which are commonly used for partially observable tasks. Moreover, we showed that these policies can work under transition and observation noise and they can generalize to novel environment with unseen objects. When transferred to real world, the policies were able to work without domain randomization or fine tuning. For future work, we would like to explore other manipulation tasks that can take advantage of fusion of tactile and visual modalities.

REFERENCES

[1] T. Keleştemur, C. Keil, J. P. Whitney, R. Platt, and T. Padır, "Learning bayes filter models for tactile localization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9253–9258.

[2] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Pomdps for robotic tasks with mixed observability." in *Robotics: Science and systems*, vol. 5, 2009, p. 4.

[3] K. Hsiao, L. P. Kaelbling, and T. Lozano-Perez, "Grasping pomdps," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4685–4692.

[4] K. Gadeyne and H. Bruyninckx, "Markov techniques for object localization with force-controlled robots," in *10th Int'l Conf. on Advanced Robotics*, 2001.

[5] S. R. Chhatpar and M. S. Branicky, "Particle filtering for localization in robotic assemblies with position uncertainty," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3610–3617.

[6] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 707–714.

[7] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dexterous manipulation," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4302–4308.

[8] S. Javdani, M. Klingensmith, J. A. Bagnell, N. S. Pollard, and S. S. Srinivasa, "Efficient touch based localization through submodularity," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1828–1835.

[9] B. Saund, S. Chen, and R. Simmons, "Touch based localization of parts for high precision manufacturing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 378–385.

[10] J. Liang, A. Handa, K. Van Wyk, V. Makoviychuk, O. Kroemer, and D. Fox, "In-hand object pose tracking via contact feedback and gpu-accelerated robotic simulation," *arXiv preprint arXiv:2002.12160*, 2020.

[11] M. Pfanne, M. Chalon, F. Stulp, and A. Albu-Schäffer, "Fusing joint measurements and visual features for in-hand object pose estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3497–3504, 2018.

[12] A. S. Lambert, M. Mukadam, B. Sundaralingam, N. Ratliff, B. Boots, and D. Fox, "Joint inference of kinematic and force trajectories with visuo-tactile sensing," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3165–3171.

[13] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[14] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine, "Omnitact: A multi-directional high resolution touch sensor," *arXiv preprint arXiv:2003.06965*, 2020.

[15] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using gelsight tactile sensing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3988–3993.

[16] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4000–4007.

[17] R. Platt, F. Permenter, and J. Pfeiffer, "Using bayesian filtering to localize flexible materials during manipulation," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 586–598, 2011.

[18] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Localizing the object contact through matching tactile features with visual map," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3903–3908.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[21] R. Jonschkowski and O. Brock, "End-to-end learnable histogram filters," in *Workshop on Deep Learning for Action and Interaction at NIPS*, December 2016.

[22] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: End-to-end learning with algorithmic priors," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[23] P. Karkus, D. Hsu, and W. S. Lee, "Particle filter networks with application to visual localization," *arXiv preprint arXiv:1805.08975*, 2018.

[24] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop kf: Learning discriminative deterministic state estimators," in *Advances in Neural Information Processing Systems*, 2016, pp. 4376–4384.

[25] M. Karl, M. Soelch, J. Bayer, and P. Van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," *arXiv preprint arXiv:1605.06432*, 2016.

[26] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *Advances in neural information processing systems*, 2015, pp. 2746–2754.

[27] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *arXiv preprint arXiv:1507.06527*, 2015.

[28] 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.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, *et al.*, "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," *arXiv preprint arXiv:1802.01561*, 2018.

[31] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[32] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2555–2565.

[33] P. Karkus, D. Hsu, and W. S. Lee, "Qmdp-net: Deep learning for planning under partial observability," in *Advances in Neural Information Processing Systems*, 2017, pp. 4694–4704.

[34] L. P. Kaelbling, "Learning to achieve goals," in *IJCAI*. Citeseer, 1993, pp. 1094–1099.

[35] D. S. Chaplot, E. Parisotto, and R. Salakhutdinov, "Active neural localization," in *International Conference on Learning Representations*, 2018.

[36] S. K. Gottipati, K. Seo, D. Bhatt, V. Mai, K. Murthy, and L. Paull, "Deep active localization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4394–4401, 2019.

[37] F. Wirnshofer, P. S. Schmitt, G. von Wichert, and W. Burgard, "Controlling Contact-Rich Manipulation Under Partial Observability," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020.

[38] 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.

[39] E. Schwarm, K. M. Gravesmill, and J. P. Whitney, "A floating-piston hydrostatic linear actuator and remote-direct-drive 2-dof gripper," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 7562–7568.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

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

[42] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[43] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[44] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3dnet: Large-scale object class recognition from cad models," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 5384–5391.