AcTExplore: Active Tactile Exploration on Unknown Objects

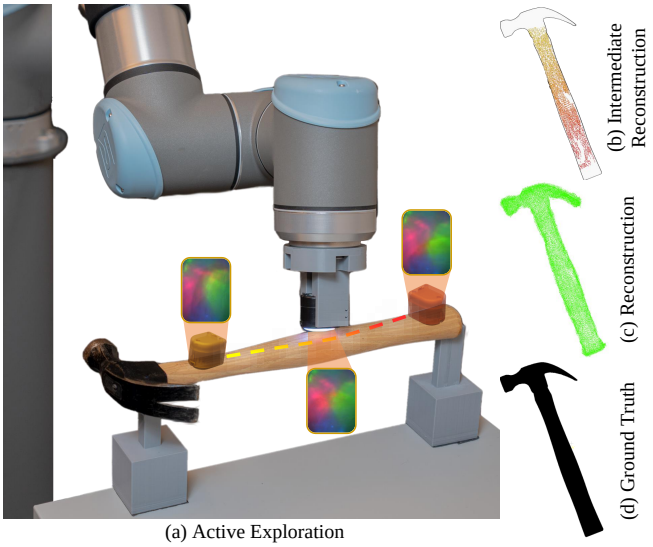
Amir-Hossein Shahidzadeh, Seong Jong Yoo, Pavan Mantripragada,
Chahat Deep Singh, Cornelia Fermüller, Yiannis Aloimonos

***Abstract*— Tactile exploration plays a crucial role in understanding object structures for fundamental robotics tasks such as grasping and manipulation. However, efficiently exploring such objects using tactile sensors is challenging, primarily due to the large-scale unknown environments and limited sensing coverage of these sensors. To this end, we present AcTExplore, an active tactile exploration method driven by reinforcement learning for object reconstruction at scales that automatically explores the object surfaces in a limited number of steps. Through sufficient exploration, our algorithm incrementally collects tactile data and reconstructs 3D shapes of the objects as well, which can serve as a representation for higher-level downstream tasks. Our method achieves an average of 95.97% IoU coverage on unseen YCB objects while just being trained on primitive shapes.**

I. Introduction

Human perception of the environment is a multifaceted process that involves multi-sensor modalities, including vision, audition, haptic, and *proprioception*. While deep learning has made significant progress in visual perception, conventional vision-only models have limitations compared to human perception. Humans excel at perceiving objects in challenging environments, utilizing their multi-sensor inputs [36], [8] such as eye for visual properties and skin for tactile sensing which is essential to characterize physical properties such as texture, stiffness, temperature, and contour [53], [3], [21]. Thus, Vision and tactile sensation have distinct roles in Scene perception, each with unique requirements. Vision relies on direct line-of-sight unobstructed views, whereas tactile sensation only necessitates physical contact, enabling perception in challenging scenarios like occluded or dark environments when vision is limited. This distinction underscores the value of tactile sensing in Scene perception, motivating the development of AcTExplore. Our goal is to maximizes contact with the object's surface during exploration, thereby fully utilizing the benefits of tactile sensation that aligns with the capabilities of humans and other living beings to bridge the gap between machine and human perception.

The human skin, our largest organ, allows us to perceive contact with the external world. This has prompted behavioral studies [20], [4], [33] which investigate human manipulation skills, where the significance of tactile sensing becomes evident. Similarly, the successful automation of robotic manipulative tasks heavily relies on their perceptual capabilities. Consequently, a multitude of tactile sensors have been developing for robotic applications, encompassing optical-based sensors like DIGIT [25] and GelSight [56], which demonstrate remarkable proficiency in discerning skin deformation [50]. Conversely, bio-inspired electrode-based sensors such as SynTouch BioTac [51] necessitate extensive post-processing and finite-element modeling to accurately represent skin deformation, as studied extensively in [38], [37].



(a) Active Exploration

Fig. 1: Reconstruction of a hammer: (a) demonstrates the tactile sensor trajectory in 3D. (b) illustrates the corresponding intermediate tactile readings on the hammer's surface. Note that we use the yellow-to-red gradient to denote the time. After thorough tactile exploration, we obtain a complete reconstruction of the object (c), indicating that the tactile sensor can cover the entire object's surface through our active strategy.

Tactile sensor outputs a detailed local perspective of objects that gives them a unique role in surface exploration. However, predicting future actions (moves) based on a single touch for exploration of the entire object's shape presents a challenging task. The challenge involves two main aspects. Firstly, the agent can stuck in repetitive movement loops and revisit areas that have already been explored. Secondly, since the agent lacks knowledge about the object's shape, it is hard to infer which action will maintain the touch to avoid exploring the free space while persistently exploring the object. Achieving a suitable trade-off between exploring new areas and maintaining contact necessitates an active policy that considers a history of tactile readings along with the

All authors are associated with the Perception and Robotics Group at the University of Maryland, College Park. The support by Brin Family Foundation, the Northrop Grumman Mission Systems University Research Program, ONR under grant award N00014-17-1-2622 and National Science Foundation under grant BCS 1824198 are gratefully acknowledged.

The supplementary material and PDF are available at http://prg.cs.umd.edu/AcTExplore

traversed trajectory of the sensor. Our active policy facilitates the sensor to avoid *Non-Exploratory Scenarios* enabling us to explore the objects within a limited number of actions. In **AcTExplore**, we address the aforementioned challenges by formulating it as a Partially Observable Markov Decision Process (POMDP) where the policy is provided with only a recent trajectory rather than a full interaction history.

To this end, we propose **AcTExplore**, an active method for tactile exploration that utilizes a deep reinforcement learning, and the **core contributions** are given as follows: (a) *Exploring Object's Surface* with minimal actions without limiting the approach to a specific distribution of objects. We achieve this by training the agent to learn dexterous movements from fundamental actions on primitive shapes (cube, sphere, etc.). Remarkably, as demonstrated in the experiments (Sec. IV), the learned behavior extends to unseen objects. (b) Introducing *Temporal Tactile Sensing* in state representation (Sec. III-B) to enable Short-Term Memory (STM) on taxels (tactile receptors), Inspired by various neurological and behavioral studies [26], [22]. (c) Proposing a curiosity-driven *Active Exploration Algorithm* for 3D Reconstruction at scale that can be integrated to various high-level tasks in future, such as Grasp Pose Refinement [13], Scene Perception [27], [48]

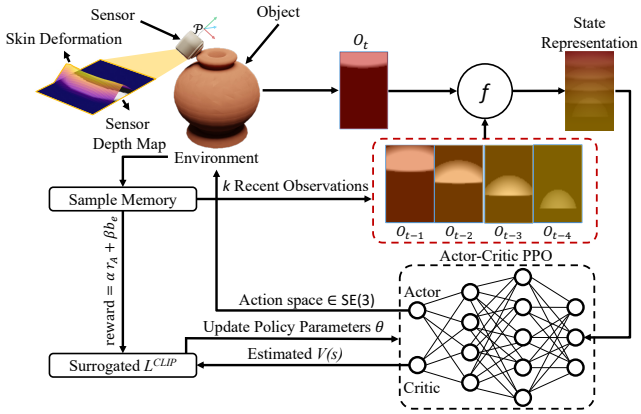


Fig. 2: **Overview:** This figure illustrates the key steps and components of AcTExplore in a scenario where the sensor moves upward along the jar's edge. We employed Temporal Tactile Averaging for state representation f (Sec. III-B) to encode consecutive observations, enabling the perception of movement on sensor, vital for learning dexterous actions. We also incorporate an Upper Confidence Bound (UCB) exploration as a bonus to encourage effective exploration.

II. Related Works

a) Active SLAM: involves traversing an unknown environment while simultaneously localizing and constructing a map [10], [6], [9], [11]. In traditional SLAM, revisiting the marked areas is beneficial for correcting estimated localization errors [31], [44]. However, our objective is to achieve an efficient exploration pipeline that minimizes revisits. Active SLAM is generally formulated as a Partially Observable Markov Decision Process (POMDPs) [44], with various reward function formulations, including curiosity [6], coverage [12], entropy-based [5], and etc. Unlike the conventional active SLAM setting, which operates in unknown 2D spaces, our work focuses on the exploration of a 3D workspace with limited sensing space which leads to ambiguity [2].

b) Deep reinforcement learning in exploration: Recent advances in computing power and physics-based simulators have boosted research in virtual navigation and exploration. For instance, [35] trained an Asynchronous Advantage Actor-Critic (A3C) agent in a 3D maze, incorporating long short-term memory (LSTM) to provide the memory capability [46], [17]. Furthermore, [43] tackled the robot exploration problem using the D-optimality criterion as an intrinsic reward which significantly accelerated the training process. In our problem, we incorporate an exploration bonus as a reward function. This approach provides the agent with incentives to explore undiscovered state and action pairs, leading to more sample efficient algorithm [1], [19].

c) Tactile applications: Tactile information plays a crucial role in human perception, encompassing tasks from object manipulation to emotional expressions. As a result, tactile sensors have been employed in various applications [28], [50], [41]. Especially, when robots manipulate deformable objects, tactile sensors provide meaningful information that enhances system robustness alongside vision sensors [42]. Furthermore, tactile information has been used to estimate the pose of the objects [2], [49], [57] or the relative pose of gripper for object handling [24]. Similar to our work, tactile sensors have been employed to identify unknown objects [32], [29], [18] or reconstruct it [40]. For instance, [54] designed a tactile object classification pipeline that actively collects tactile information while exploring the object. The closest previous studies that addresses the 3D reconstruction of unknown objects through information-theoretic exploration were [30], [15], [55] from which some were primarily evaluated in simulation and some can work for simple objects when compared to real world YCB [7] objects, which present challenges in terms of collision avoidance during planning when object's geometry is unknown. Some other studies have focused on shape reconstruction, specifically handling missed segments individually [47], [14]. However, their primary focus was on shape completion using a passive exploration algorithm. In contrast, our work addresses the challenges of 3D object active exploration in both simulation and real-world, facilitating the reconstruction process by exploring the object in limited trials.

III. Method

In AcTExplore, we consider a tactile sensor mounted on a robotic arm end-effector, along with access to the end-effector pose $\mathcal{P}_t \in SE(3)$ from forward kinematics (FK) interacting with an unknown fixed 3D object. The goal is to navigate across the entire surface of an unknown object within a limited workspace to collect tactile data for the above-mentioned reasons in Sec. I. At time t, the model will utilize consecutive tactile data $\{O_t, O_{t-1}, \ldots, O_{t-(k-1)}\}$ to generate exploratory action $a_t \in SE(3)$ based on k recent observations. The effectiveness of the Completeness process heavily relies on the robot's exploration algorithm.

An exploration policy, denoted as π_θ (referred to as the explorer) determines the next action, $\arg\max_{a_t} \pi_\theta(a_t|s_t)$, based on state s_t to maximize cumulative reward (Sec. III-D) which serves as an estimate of coverage over an

unknown object's shape. Therefore, the problem formulation of AcTExplore encompasses four key components:

A. Observation Space

At each time step t, the sensor observes skin deformation as an image (Fig. 2) that can be converted into depth map of skin [25], [50]. We'll denote this depth image as observation space $O_t \in R^{H \times W}$ which corresponds to the deformation that the sensor has at pose $\mathcal{P}_t$.

B. State Representation

The state serves as the sole input of the explorer so it has to be sufficiently informative, enabling the model to generate exploratory actions. Let s_t be the state which serves as the input for π_θ at time t. Considering the possibility that multiple optimal actions correspond to the observation O_t at finger pose $\mathcal{P}_t$, it is advantageous to construct a state representation s_t that incorporates short-term memory. This can be achieved by using a sequence of k consecutive observations $\{O_t, O_{t-1}, \ldots, O_{t-(k-1)}\}$ to encapsulate the complexity of the state, enabling the policy to generate the appropriate action for long-horizon exploration. Therefore, $s_t = f(O_t, O_{t-1}, \ldots, O_{t-(k-1)})$, where f can be any function, connecting spatio-temporal information on sensor like optical flow which is computationally costly to generate on the fly. The purpose of this function is to reduce the dimensionality of s_t while extracting critical features for the state representation [39] which are not feasible to infer from a single observation. This is particularly valuable for learning high-level, complex actions that require a wider view. We conduct extensive experiments investigating the effectiveness of various state representations in Section IV. In Fig. 2, we visualized the resulting tactile readings and our proposed functions f:

$$\text{TTS}(O_t, O_{t-1}, ..., O_{t-(k-1)}) : O_t \parallel O_{t-1} \parallel \ldots \parallel O_{t-(k-1)}$$
(Temporal Tactile Stacking)

$$\text{TTA}(O_t, O_{t-1}, ..., O_{t-(k-1)}) : \sum_{i=0}^{k-1} \alpha_i O_{t-i}$$
(Temporal Tactile Averaging)

in which α_i are decreasing and $\sum_{i=0}^{k-1} \alpha_i = 1$ so the most recent reading O_t is the most effective observation in s_t and others will be averaged respectively.

C. Action Space

To efficiently explore the complex geometry of 3D objects, we enable the finger to move in a 6-degree-of-freedom (6DOF) action space denoted as $A \in SE(3)$. In this action space, we consider small incremental translations (x, y, z) and rotations (γ, θ, ψ) around the workspace frame, with the bottom of the finger serving as the reference point. The model selects one of the dimensions $(x, y, z, \gamma, \theta, \psi)$ and either increases or decreases it by the specified step. This results in a total of 12 possible actions within the action space. This action space is chosen to facilitate the control of the sensor under kinematic and collision constraints of the arm in both simulation and real world.

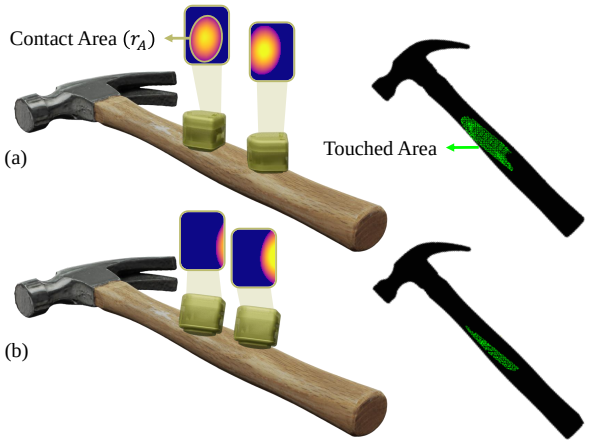


Fig. 3: Depth readings of DIGIT sensor sliding on a hammer (a): Sensor aligned with object's surface, receiving more depth information and moving stably. (b): Misaligned rotation leads to a higher probability of losing contact in future steps.

While this action space is capable of facilitating exploration and interaction, it can be enhanced by adding an action that allows the finger to return to the last touching location. This additional action serves the purpose of touch recovery and addressing Non-Exploratory scenarios where contact with the object may be temporarily lost. By including this *Touch Recovery action* (a_{TR}), the robot can reestablish contact and give up on the trajectories that aren't worthy of further exploration. It will also guarantee that the model never strays too far from the object, as it learns to perform this action after a certain number of steps without any touch. Therefore, the action space consists of $12 + 1$ actions in total.

D. Reward

We can break down the exploration objective into easier sub-goals. Specifically, In this work, we want to maximize the coverage of objects through the trajectory of the tactile sensor. To accomplish this, the reward function can be divided into two main sub-goals:

1) **Contact Area Reward** measures the area of contact between the robot's finger and the object's surface. The rationale behind this is that a larger contact area corresponds to a greater amount of information being gathered from the object as illustrated in Figure 3. In other words, this reward will encourage actions that align the tactile sensing area of the finger with the object's surface.

2) **Exploration Bonus:** To further encourage the agent to explore the workspace from the global point of view and inspired by [1], [19], we have introduced a memory mechanism to track all the trajectories taken by the agent. This memory allows us to keep a count of the number of times the agent has performed a particular action a in a specific pose $\mathcal{P}$, denoted as $N(\mathcal{P}, a)$. Even so, using $N(\mathcal{P}, a)$ directly for exploration is not practical, because most of the states would have $N(\mathcal{P}, a) = 0$, especially considering that the workspace is often continuous or high-dimensional. So instead we define

$\hat{N}(\mathcal{P}, a)$ as the count of the number of close poses (Sec. SI-C-1) in trajectory history. By having access to $\hat{N}(\mathcal{P}, a)$ for each pose-action pair, we can incorporate a bonus term $b_e = \frac{1}{\sqrt{\hat{N}(\mathcal{P},a)}}$ into the area reward r_A in case that the $r_A > 0$ which indicates that the sensor is touching the object. The bonus term b_e is a term that incentivizes exploration and curiosity throughout the trajectory. By incorporating this bonus, actions that have been infrequently taken in the past on a specific $\mathcal{P}$ are rewarded more prominently, encouraging the agent to explore less-visited state-action pairs. Furthermore, the agent is guided toward more comprehensive exploration and exhibits a tendency to venture into uncharted regions of the workspace. This aids in mitigating the impact of sub-optimal local optima and fosters a broader understanding of the environment, resulting in improved convergence and enhanced exploration behavior as discussed in [19].

3) **Penalties** There are two possible trivial local optima scenarios that the agent might learn to maximize the area reward and exploration bonus without exploring the object in long-horizon. To address this issue, we'll define necessary negative rewards: **Revisit Penalty (P_{rev}) :** In order to prevent the agent from learning policies that involve revisiting recently visited locations we'll construct a short-term memory $\mathcal{D} = \{\mathcal{P}_i\}_{i=t-m}^{t}$ of m recent interactions by time t. If an action leads the sensor to revisit a location already present in its short-term history $\mathcal{D}$, a revisit penalty $P_{rev} < 0$ will be imposed to discourage such trajectories that no novel information can be obtained. **Touch Recovery Penalty (P_{TR}):** One possible scenario that may not be covered by P_{rev} is when the sensor moves freely in space without making contact with the object for more than m steps which ($|\mathcal{D}|$), and then performs a touch recovery action, which has a positive area reward ($r_A > 0$). To prevent such scenarios, we introduce a negative reward P_{TR} each time the agent selects the touch recovery action (a_{TR}). However, despite this negative reward, the agent still tends to choose the touch recovery action due to the high value of $V(s_{t+1})$ associated with recovery actions. Additionally, we can control the number of actions without touch by adjusting the P_{TR}.

By penalizing non-exploratory scenarios, the agent is incentivized to explore new areas, mitigating the risk of getting stuck in sub-optimal loops and performing dexterous actions on a long horizon. (Please refer to Sec. SI-B-1[1] for further discussion.)

With all of these considerations in mind, the reward function is formulated as follows:

$$r(s_t, a_t) = \begin{cases} \alpha r_A + \frac{\beta}{\sqrt{\hat{N}(\mathcal{P}_t, a_t)}}, & \text{if } r_A > 0 \text{ and } \mathcal{P}_{t+1} \notin \mathcal{D} \\ P_{rev}, & \text{if } r_A > 0 \text{ and } \mathcal{P}_{t+1} \in \mathcal{D} \\ P_{TR}, & \text{if } a_t = a_{TR} \text{(touch recovery)} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where r_A represents the reward based on the area of contact between the sensor and the object's surface at time t. P_{rev} denotes the penalty term applied to actions that lead to previously visited locations in $\mathcal{D}$. $\hat{N}(\mathcal{P}_t, a_t)$ signifies number of times the agent has performed action a_t in pose $\mathcal{P}_t$ by time t, which is used to calculate the exploration bonus term.

By combining these components within the reward function, we aim to achieve the balance between contact area maximization (i.e., *exploitation*), and avoidance of non-exploratory scenarios (i.e., *exploration*) which addresses the mentioned difficulties for training in an unknown environment.

In order to compute variance-reduced advantage/value function estimators, AcTExplore utilizes a modified Proximal Policy Optimization (PPO) algorithm by modeling the exploration objective as an intrinsic auxiliary reward and enriching the state with temporal representation. We summarized our overall method in Alg. 1. The environment is initialized with an object to be explored and a tactile sensor to actively move on the object's surface and collect observations. We initialize the workspace boundaries around the object with the object's center serving as the global frame. In each episode, the sensor starts moving toward the workspace center until the first touch when the algorithm starts training. The agent continues to interact with the object to learn optimal actions through multi-objective reward **AMB** (Sec. III-D) that estimates coverage that is not available during training on unknown objects.

Algorithm 1: ACTEXPLORE TRAINING

for *episode = 1,2, ...* **do**
- $\mathcal{D} \leftarrow$ List of size m
- $\mathcal{P}_0 \leftarrow$ First touch pose
- $O_0 \leftarrow$ Tactile sensor reading at $\mathcal{P}_0$
- $\hat{N}(p, a) \leftarrow 0$ for all $(p, a) \in \mathcal{P} \times \mathcal{A}$
- **for** *t = 0, 1, 2, …, T-1* **do**
  - $s_t \leftarrow f(\{O_i\}_{i=t-\min(k,t)}^{t})$
  - $a_t \leftarrow \arg\max_{a'} \pi_\theta(a'|s_t)$
  - $\mathcal{P}_{t+1}, O_{t+1} \leftarrow$ tactileSensor.step(a_t)
  - $\hat{N}(\mathcal{P}_t, a_t) \leftarrow \hat{N}_t(\mathcal{P}_t, a_t) + 1$
  - $r_A \leftarrow$ nonZeroCount(O_t) / size(O_t)
  - $r \leftarrow 0$
  - **if** $a_t = a_{TR}$ **then**
    - $r \leftarrow P_{TR}$ `// touch recovery`
  - **else if** $\mathcal{P}_{t+1} \in \mathcal{D}$ **then**
    - $r \leftarrow P_{rev}$ `// revisit`
  - **else if** $r_A > 0$ **then**
    - $b_e \leftarrow \frac{1}{\sqrt{\hat{N}_t(\mathcal{P}_t, a_t)}}$
    - $r \leftarrow \alpha r_A + \beta b_e$
  - $\mathcal{D}$.add($\mathcal{P}_{t+1}$)
  - $\delta_t \leftarrow r + \gamma V(s_{t+1}) - V(s_t)$
- **end**
- Compute advantages $\hat{A}_{i\in[T]} : \sum_{j=i}^{T-1} \gamma^{j-i}\delta_{i+j}$
- $\theta \leftarrow$ Optimize surrogate $L^{CLIP}(\theta, \hat{A}_{[T]})$

end

IV. EXPERIMENTS

In this section, we evaluate and analyze our method AcTExplore with various rewards and states on zero-shot (*unseen*) objects. In addition, we validate our method on over 400

[1]The supplementary material is available at http://prg.cs.umd.edu/AcTExplore

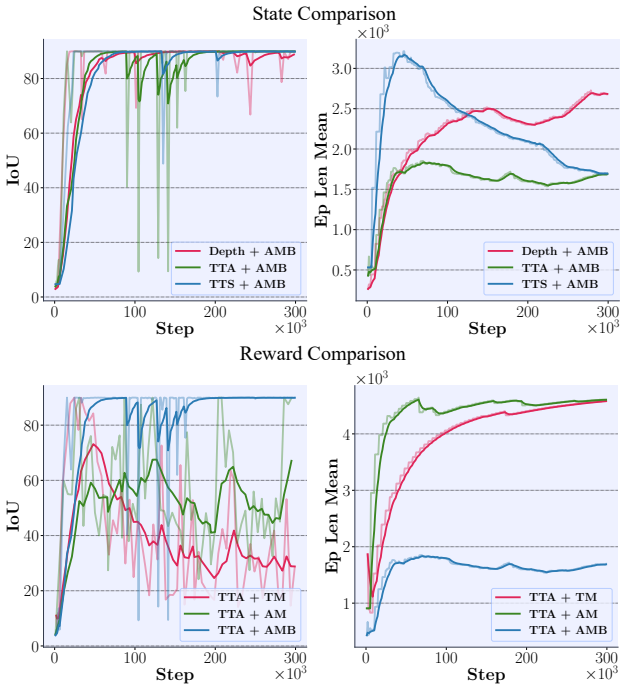


Fig. 4: **Training results:** The left two graphs compare state representation while using AMB as reward function, and the right two graphs showcase different reward settings while using TTA as state representation. Note that episodes terminate either when the agent surpasses 90% IoU, reaches the horizon steps, or reaches the workspace boundaries.

quantitative and qualitative experiments in the reconstruction of unknown objects with varying complexities. In our experiments, *we use reconstruction accuracy as the metric for tactile exploration with a limited number of steps as it represents the AcTExplore exploration potential.*

A. Experimental Setup

We employ TACTO [52], [16] to simulate tactile sensor skin deformation during object interactions and modified PPO from StableBaselines3 [45] in AcTExplore. The TACTO simulator is calibrated with real-sensor data to ensure Sim-to-Real generalization. It generates depth map images from real-world signals, serving as our observation (O). We train the agent only with primitive objects – sphere and cube – for 300K steps. These primitive objects are selected as they represent a broad range of object shapes with the sphere having curvature and the cube having sharp edges, corners, and flat surfaces. To assess the model's performance, we evaluate it on YCB objects that were not encountered during training time. This evaluation demonstrates the efficacy of training with primitives which exhibit strong generalization capabilities for objects with realistic textures(Fig. S2). For the termination condition, each episode either spans 5000 steps(Sec. SI-A) or concludes once the Intersection over Union (IoU) metric exceeds 90%, or when the sensor moves beyond the workspace boundaries. This strategy is adopted to optimize training time. In Table II, we show that these termination conditions won't limit the IoU performance during the test time as our methods reached above 90%.

B. Baselines Configuration

To evaluate the efficacy of each component, we have established a collection of baselines for three different state representations and three different reward functions, summarized at Table I.

TABLE I: Baseline Formulations. **TTA**: Temporal Tactile Averaging, **TTS**: Temporal Tactile Stacking (concatenation is denoted as $\|$), **TM**: binary Touch indicator ($\mathbb{I}(\cdot)$) + short Memory, **AM**: contact Area + short Memory, **AMB**: contact Area + short Memory + UCB Bonus

State	Depth	TTA	TTS
	O_t	$\sum_{i=0}^{k-1} \alpha_i O_{t-i}$	$O_t \,\|\| \ldots \|\| \, O_{t-(k-1)}$
Reward	TM	AM	AMB
	$\mathbb{I}(O_t)$	$r_A(O_t)$	$\alpha r_A(O_t) + \frac{\beta}{\sqrt{\hat{N}(\mathcal{P}_t, a_t)}}$

C. Analysis & Discussion

a) State Comparison: We perform a comparison of different state representations using the same reward function (AMB), considering both representations with and without temporal information. This analysis highlights the influential impact of temporal information on learning dexterous and high-level actions. As shown in Fig. 4, all state representations achieve a specified IoU during training. However, the state representations incorporating temporal information demonstrate higher stability, consistently reaching the 90% IoU objective after 200K steps. In contrast, the depth-only representation struggles to maintain the IoU objective and will be outperformed by temporal representations in Table II. Furthermore, when considering the number of steps required

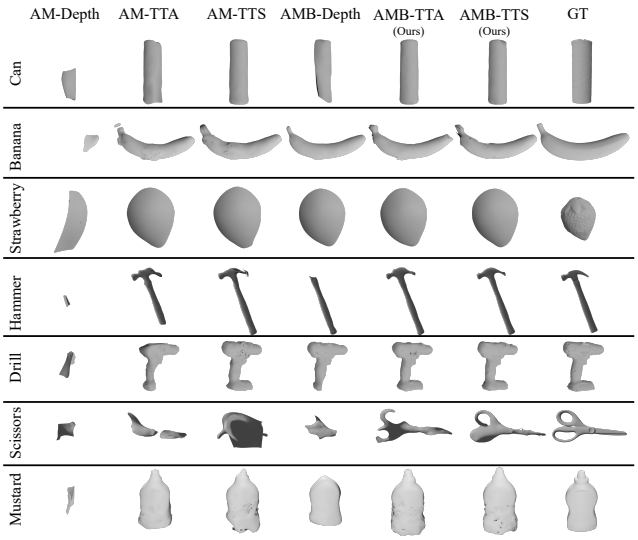


Fig. 5: **Qualitative results on unseen YCB objects** with different state and reward settings. From active tactile exploration, we obtain point cloud data on the object's surface. To generate mesh from the collected point cloud, we apply Poisson surface reconstruction algorithm [23]. Further experiments are provided in supplementary materials.

TABLE II: **Quantitative results on unseen YCB objects:** The table presents IoU and Chamfer-L1 distance (cm) [34] values of the predicted meshes and ground-truth meshes. The surface area is listed below each object's name as a severity metric. The details of metrics, confidence intervals, and step counts are mentioned in the supplementary material.

Models \ Objects		Can (616 cm^2)	Banana (216 cm^2)	Strawberry (68 cm^2)	Hammer (410 cm^2)	Drill (591 cm^2)	Scissors (165.48 cm^2)	Mustard (454.54 cm^2)
		IoU ↑ (Chamfer-L_1 ↓)						
TM	depth	31.93 (2.66)	11.11 (7.52)	83.60 (0.44)	32.78 (1.86)	19.19 (4.1)	24.29 (8.15)	10.07 (4.07)
	TTA	17.60 (3.57)	6.03 (9.03)	41.0 (1.23)	14.85 (6.94)	28.15 (3.99)	14.17 (4.98)	19.94 (3.22)
	TTS	15.93 (5.22)	18.23 (5.48)	57.89 (0.88)	28.66 (2.47)	15.5 (3.97)	11.26(4.97)	14.55(2.95)
AM	depth	11.59 (5.49)	10.22 (6.84)	47.33 (1.16)	5.07 (7.69)	9.49 (4.03)	5.11(6.78)	11.04(5.16)
	TTA	72.70 (0.56)	97.70 (0.35)	**100** (0.28)	79.80 (0.82)	57.58 (1.43)	41.77 (2.87)	71.72 (0.80)
	TTS	**98.25** (0.22)	**100** (0.34)	**100** (0.31)	88.22 (0.44)	99.02 (0.37)	28.37 (2.38)	87.13 (0.59)
AMB	depth	41.45 (1.42)	98.64 (**0.25**)	**100 (0.23)**	61.42 (1.17)	79.68 (0.95)	31.99 (3.2)	65.74 (0.9)
	depth+LSTM	88.54 (0.3)	99.96 (0.28)	**100** (0.24)	87.54 (0.49)	92.81 (0.36)	29.83 (0.58)	88.33 (0.36)
	TTA (ours)	89.6 (0.29)	**100** (0.33)	**100** (0.25)	**98.22** (0.29)	98.85 (0.32)	67.02 (0.87)	**95.91** (0.51)
	TTS (ours)	97.45 (**0.20**)	**100** (0.3)	**100** (0.25)	96.96 (**0.28**)	**99.74 (0.31)**	**74.62** (**0.61**)	95.02 (**0.49**)

to achieve the IoU objective, TTS training takes longer than TTA as $s_t^{TTS} \in R^{k\times H\times W}$ is k times bigger (Eq. Temporal Tactile Stacking) than $s_t^{TTA} \in R^{H\times W}$ which is averaging observations rather than stacking them. However, in our experiments Figure 5, we witnessed that both TTA and TTS are quite competitive but TTS performs better on long objects while TTA works better on shapes with more complexity.

b) Reward Comparison: In our pursuit of efficient exploration, we tried various reward functions as described in Sec. IV-B. During training, we plotted the IoU and episode length until termination in Fig. 4. Notably, AMB reward function outperformed the others, satisfying the IoU objective through encouraging exploration of less visited locations. In contrast, TM and AM are not capable of making use of environmental feedback as much as AMB can. This limitation arises from TM and AM's deprivation of long-horizon history, which hampers their capacity to gather sufficient information through intrinsic rewards. As a result, AMB is better equipped to leverage environment feedback $\left(\frac{1}{\hat{N}(\mathcal{P},a)}\right)$ effectively for improved exploration and sample efficiency. However, AM is outperforming TM as it utilizes contact area information and still has the ability to align the sensor's sensing area with the object surface to collect more information and maintain a reliable touch for future actions. Indeed, The disparity between TM and AM can also be understood as the distinction between using a touch sensor versus a tactile sensor for exploring an object.

c) Real-World Setup: In our experimental configuration, we employ a UR10 arm to manipulate the 6D pose of the DIGIT (Fig. 1). This control is achieved by transforming changes in the DIGIT's frame into a set of joint trajectories via inverse kinematics. The resulting trajectories are executed only when they are free from self-collisions and within the defined workspace around the object. In instances where an invalid trajectory is generated, an alternative action is chosen based on the PPO's advantage values.

Unlike simulations, where consecutive action executions while in contact with the object have minimal impact, our real-world implementation introduces significant shearing of the sensing surface. To ensure the safe execution of actions generated by our policy, we have adopted a strategy of lifting the DIGIT in the normal direction of the contact after each contact event. This strategy remains well-founded due to the consistent alignment of our sensor surface with the object's surface by our policy and does not compromise its general applicability. Our method successfully transferred to real-world experiments with no further fine-tuning required.

d) Limitations: The current formulation of our method has certain limitations. First, it assumes a moving sensor relative to a **fixed-pose** rigid object. Although AcTExplore is not restricted by object shape, but it is designed to keep the sensor close to recent touching poses. This could pose challenges in environments with disconnected components. Workspace splitting can be a potential solution to address this problem. Second, the sensor exhibits a small depth bias in the simulation resulting in larger reconstructions. While generally negligible, this bias becomes dominant when handling objects roughly the same size as the sensor, such as the strawberry shown in Fig. 5.

V. Conclusion

In this work, we introduced a novel reinforcement learning method using tactile sensing that can actively explore unknown 3D objects. It addresses the need for an active exploration method to enable numerous works [13], [27], [48] to become fully automated. AcTExplore is not limited to any specific shape distributions as it has been only trained on primitive shapes to learn fundamental movements by leveraging temporal tactile information and intrinsic exploration bonuses. We demonstrated this through our experiments with various shape complexities like Drill or Hammer in both real world and simulation.

REFERENCES

[1] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, 2017.

[2] Maria Bauza, Antonia Bronars, and Alberto Rodriguez. Tac2Pose: Tactile Object Pose Estimation from the First Touch. 2022. Publisher: arXiv Version Number: 2.

[3] Wouter M. Bergmann Tiest and Astrid M. L. Kappers. The influence of visual and haptic material information on early grasping force. *Royal Society Open Science*, 6(3):181563, 2019.

[4] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364:eaat8414, 06 2019.

[5] N. Botteghi, Beril Sirmacek, R. Schulte, M. Poel, and C. Brune. REINFORCEMENT LEARNING HELPS SLAM: LEARNING TO BUILD MAPS. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 43, 2020.

[6] Nicolò Botteghi, Rob Schulte, Beril Sirmacek, Mannes Poel, and Christoph Brune. Curiosity-Driven Reinforcement Learning Agent for Mapping Unknown Indoor Environments. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:129–136, 2021. Publisher: Copernicus GmbH.

[7] Berk Çalli, Aaron Walsman, Arjun Singh, Siddhartha S. Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols. *CoRR*, abs/1502.03143, 2015.

[8] Céline Cappe, Gregor Thut, Vincenzo Romei, and Micah M. Murray. Selective integration of auditory-visual looming cues by humans. *Neuropsychologia*, 47(4):1045–1052, March 2009.

[9] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to Explore using Active Neural SLAM, April 2020. arXiv:2004.05155 [cs].

[10] Fanfei Chen, Shi Bai, Tixiao Shan, and Brendan Englot. Self-learning exploration and mapping for mobile robots via deep reinforcement learning. In *Aiaa scitech 2019 forum*, page 0396, 2019.

[11] Fanfei Chen, John D. Martin, Yewei Huang, Jinkun Wang, and Brendan Englot. Autonomous exploration under uncertainty via deep reinforcement learning on graphs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6140–6147. IEEE, 2020.

[12] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning Exploration Policies for Navigation, March 2019. arXiv:1903.01959 [cs].

[13] Cristiana de Farias, Naresh Marturi, Rustam Stolkin, and Yasemin Bekiroglu. Simultaneous tactile exploration and grasp refinement for unknown objects. *CoRR*, abs/2103.00655, 2021.

[14] Won Kyung Do and Monroe Kennedy. DenseTact: Optical Tactile Sensor for Dense Shape Reconstruction. *2022 International Conference on Robotics and Automation (ICRA)*, pages 6188–6194, May 2022.

[15] Danny Driess, Daniel Hennes, and Marc Toussaint. Active multi-contact continuous tactile exploration with gaussian process differential entropy. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7844–7850, 2019.

[16] Benjamin Ellenberger. Pybullet gymperium. `https://github.com/benelot/pybullet-gym`, 2018–2019.

[17] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: visual question answering in interactive environments. *CoRR*, abs/1712.03316, 2017.

[18] Francois Robert Hogan, Michael Jenkin, Sahand Rezaei-Shoshtari, Yogesh A. Girdhar, David Meger, and Gregory Dudek. Seeing through your skin: Recognizing objects with a novel visuotactile sensor. volume abs/2011.09552, 2020.

[19] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[20] R.S. Johansson, U. Landstro¨m, and R. Lundstro¨m. Responses of mechanoreceptive afferent units in the glabrous skin of the human hand to sinusoidal skin displacements. *Brain Research*, 244(1):17–25, 1982.

[21] Astrid M. L. Kappers. Human perception of shape from touch. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1581):3106–3114, 2011.

[22] Tobias Katus and Søren K. Andersen. Chapter 21 - the role of spatial attention in tactile short-term memory. In Pierre Jolicoeur, Christine Lefebvre, and Julio Martinez-Trujillo, editors, *Mechanisms of Sensory Working Memory*, pages 275–292. Academic Press, San Diego, 2015.

[23] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006.

[24] Tarik Kelestemur, Robert Platt, and Taskin Padir. Tactile Pose Estimation and Policy Learning for Unknown Object Manipulation. 2022. Publisher: arXiv Version Number: 1.

[25] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian H. Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *CoRR*, abs/2005.14679, 2020.

[26] Rebecca Lawson, Alexandra M. Fernandes, Pedro B. Albuquerque, and Simon Lacey. Chapter 19 - remembering touch: Using interference tasks to study tactile and haptic memory. In Pierre Jolicoeur, Christine Lefebvre, and Julio Martinez-Trujillo, editors, *Mechanisms of Sensory Working Memory*, pages 239–259. Academic Press, San Diego, 2015.

[27] Wenyu Liang, Qinyuan Ren, Xiaoqiao Chen, Junli Gao, and Yan Wu. Dexterous manoeuvre through touch in a cluttered scene. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6308–6314, 2021.

[28] Changyi Lin, Ziqi Lin, Shaoxiong Wang, and Huazhe Xu. DTact: A Vision-Based Tactile Sensor that Measures High-Resolution 3D Geometry Directly from Darkness. 2022. Publisher: arXiv Version Number: 1.

[29] Justin Lin, Roberto Calandra, and Sergey Levine. Learning to Identify Object Instances by Touch: Tactile Recognition via Multimodal Matching. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3644–3650, May 2019. ISSN: 2577-087X.

[30] Yujie Lu, Jianren Wang, and Vikash Kumar. Curiosity Driven Self-supervised Tactile Exploration of Unknown Objects. 2022. Publisher: arXiv Version Number: 1.

[31] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 11(1), 2022.

[32] Uriel Martinez-Hernandez, Nathan F. Lepora, and Tony J. Prescott. Active haptic shape recognition by intrinsic motivation with a robot hand. In *2015 IEEE World Haptics Conference (WHC)*, pages 299–304, 2015.

[33] Luca Massari, Calogero Oddo, Edoardo Sinibaldi, Renaud Detry, Joseph Bowkett, and Kalind Carpenter. Tactile sensing and control of robotic manipulator integrating fiber bragg grating strain-sensor. *Frontiers in Neurorobotics*, 13, 04 2019.

[34] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. *CoRR*, abs/1611.03673, 2016.

[36] Nicole E. Munoz and Daniel T. Blumstein. Multisensory perception in uncertain environments. *Behavioral Ecology*, 23(3):457–462, May 2012.

[37] Yashraj Narang*, Balakumar Sundaralingam*, Miles Macklin, Arsalan Mousavian, and Dieter Fox. Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections (*equal contribution). *IEEE Intl. Conf. on Robotics and Automation*, pages 6444–6451, 2021.

[38] Yashraj S. Narang, Balakumar Sundaralingam, Karl Van Wyk, Arsalan Mousavian, and Dieter Fox. Interpreting and predicting tactile signals for the syntouch biotac. *CoRR*, abs/2101.05452, 2021.

[39] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *CoRR*, abs/1808.00177, 2018.

[40] Simon Ottenhaus, Lukas Kaul, Nikolaus Vahrenkamp, and Tamim Asfour. Active tactile exploration based on cost-aware information gain maximization. *International Journal of Humanoid Robotics*, 15:1850015, 02 2018.

[41] K. Park, H. Yuk, M. Yang, J. Cho, H. Lee, and J. Kim. A biomimetic elastomeric robot skin using electrical impedance and acoustic tomography for tactile sensing. *Science Robotics*, 7(67):eabm7187,

June 2022. Publisher: American Association for the Advancement of Science.

[42] Leszek Pecyna, Siyuan Dong, and Shan Luo. Visual-Tactile Multimodality for Following Deformable Linear Objects Using Reinforcement Learning. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3987–3994, October 2022. Conference Name: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) ISBN: 9781665479271 Place: Kyoto, Japan Publisher: IEEE.

[43] Julio A. Placed and José A. Castellanos. A Deep Reinforcement Learning Approach for Active SLAM. *Applied Sciences*, 10(23):8386, January 2020. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.

[44] Julio A. Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A. Castellanos. A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers. *IEEE Transactions on Robotics*, pages 1–20, 2023. Conference Name: IEEE Transactions on Robotics.

[45] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[46] Dhruv Ramani. A short survey on memory based reinforcement learning. *ArXiv*, abs/1904.06736, 2019.

[47] Edward J. Smith, Roberto Calandra, Adriana Romero, Georgia Gkioxari, David Meger, Jitendra Malik, and Michal Drozdzal. 3d shape reconstruction from vision and touch. *CoRR*, abs/2007.03778, 2020.

[48] S. Suresh, Z. Si, J. Mangelson, W. Yuan, and M. Kaess. ShapeMap 3-D: Efficient shape mapping through dense touch and vision. In *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA*, Philadelphia, PA, USA, May 2022.

[49] Sudharshan Suresh, Maria Bauza, Kuan-Ting Yu, Joshua G. Mangelson, Alberto Rodriguez, and Michael Kaess. Tactile SLAM: Real-time inference of shape and pose from planar pushing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11322–11328, May 2021. ISSN: 2577-087X.

[50] Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. MidasTouch: Monte-Carlo inference over distributions across sliding touch. In *Proc. Conf. on Robot Learning, CoRL*, Auckland, NZ, December 2022.

[51] SynTouch. Biotac product manual. Aug 2018.

[52] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. TACTO: A fast, flexible and open-source simulator for high-resolution vision-based tactile sensors. *CoRR*, abs/2012.08456, 2020.

[53] Xuelian Wei, Baocheng Wang, Zhiyi Wu, and Zhong Lin Wang. An open-environment tactile sensing system: Toward simple and efficient material identification. *Advanced Materials*, 34(29):2203073, 2022.

[54] Jingxi Xu, Han Lin, Shuran Song, and Matei T. Ciocarlie. Tandem3d: Active tactile exploration for 3d object recognition, 2022.

[55] Zhengkun Yi, Roberto Calandra, Filipe Veiga, Herke van Hoof, Tucker Hermans, Yilei Zhang, and Jan Peters. Active tactile object exploration with gaussian processes. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4925–4930, 2016.

[56] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017.

[57] Jialiang Zhao, Maria Bauzá, and Edward H. Adelson. Fingerslam: Closed-loop unknown object localization and reconstruction from visuo-tactile feedback, 2023.

Supplementary Materials for AcTExplore: Active Tactile Exploration on Unknown Objects

***Abstract*— This document provides additional discussions, figures and experiments of main paper.**

I. Implementation Details

A. Action Space

Suppose the sensor can freely move in 3D space, then it has a full 6-DOF continuous action space. However, in order to speed up the training process we discretize the 6-DOF actions into small translations (x, y, z) and rotations (γ, θ, ψ) steps. The agent can pick one of 6-DOF to decrease or increase which either translates or rotates the sensor. Therefore the 12 action space is $A = \{\pm x, \pm y, \pm z, \pm\gamma, \pm\theta, \pm\psi\}$. The translation step (T_s) is $4mm$, while the rotation step (R_s) is 15 degrees about each axis. Furthermore, we introduce *Touch Recovery action* (a_{TR}) by saving last touch pose $\mathcal{P}_{TR}$. Note that the quantity of steps required to explore objects is contingent upon the translation and orientation step size of our action space. To provide further clarity, let's consider an example. Consider an object with a surface area of $220cm^2$. Simplifying this object to a square cube with 90% of the area, each edge's length would be approximately $5.7cm$. Given a translation size of $4mm$, it necessitates about 206 actions for optimal exploration of each facet. A rotation of 15 degrees necessitates 6 actions to transition between facets. Therefore, exploring a cube of theoretically entails 1260 actions, considering our action step size. Now, if we apply this concept to the YCB's banana, which has a comparable surface area($216cm^2$) but is more intricate than a cube and necessitates additional rotations, the TTS-AMB requires 1631 actions, contrasting with the 1260 actions needed for the cube which seems reasonable when the object is curved and cylindrical and takes more rotation actions.

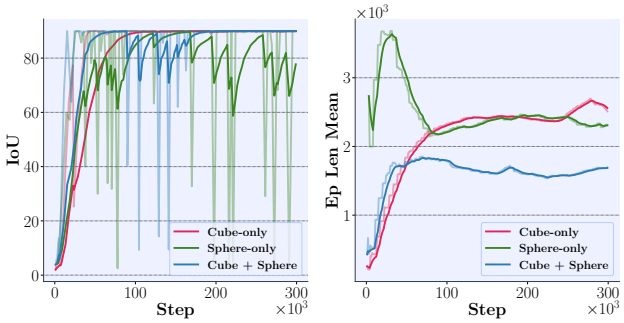


Fig. 1: **Ablation study of training primitives:** We trained AMB-TTS with cube-only and sphere-only setting as well.

B. Reward

1) Hyperparameters Tuning: Our reward function encompasses several hyperparameters, the effects of which and tuning methodologies are expounded in this section. Please note that r_A and b_e are normalized in range of $[0, 1]$, thus for tuning α and β which are designed to regularize r_A, and b_e respectively, we have tried various values, maintaining constraint $\alpha + \beta = 1$, to ensure $r(s_t, a_t) \in [0, 1]$. Our observations revealed large values of α led to learning policies that moves the agent in a loop which is bigger than short-term memory size $|\mathcal{D}| = m$ as it would receive P_{rev} in smaller loops where the required actions are less than m. Conversely, when β is too large the agent learn policies where the agent failed to align its sensing area with objects. In consideration of these factors and the distributions of r_A and b_e, we determined $\alpha = 0.15$ and $b_e = 0.85$ to effectively address the outlined issues. Regarding the tuning of P_{rev}, it is pertinent to note that its magnitude should be substantial enough to prohibit bad scenarios like loop and non-exploratory trajectories. P_{rev} has a direct interplay with m as it applies solely when the new pose $\mathcal{P}_{t+1} \in \mathcal{D}$ so with with an empirically established $m = 20$, $P_{rev} = -0.03$ results in the favorable behavior. P_{TR}'s role is to discourage the model from selecting the touch recovery action which has a positive reward as it'll touch the object's surface where $(r_A > 0)$. Furthermore, it's actually regulating the number of exploratory actions without touch as the agent is sacrificing the positive rewards of touching poses near the current pose for opting to explore surfaces that may not be directly connected or proximate to the previous pose. Finally, by choosing $P_{TR} = -0.2$, all the mentioned issues will be mitigated. To tune P_{TR}, we recommend first tuning the other hyperparameters with 12 actions(without touch recovery action) and subsequently determining the appropriate value for P_{TR} based on the complexity of the environment. The TTA representation also requires some regularizer parameters α_i which are generated from

$$\alpha_i = \frac{1 + \frac{i}{\lambda}}{\sum_{k=0}^{m} 1 + \frac{k}{\lambda}} \tag{1}$$

that satisfies $\sum_{i=0}^{m} \alpha_i = 1$ and will generate the biggest weight for the most recent observation which corresponds to α_m. In our experiments, $\lambda = 50$ results in the expected behavior from TTA.

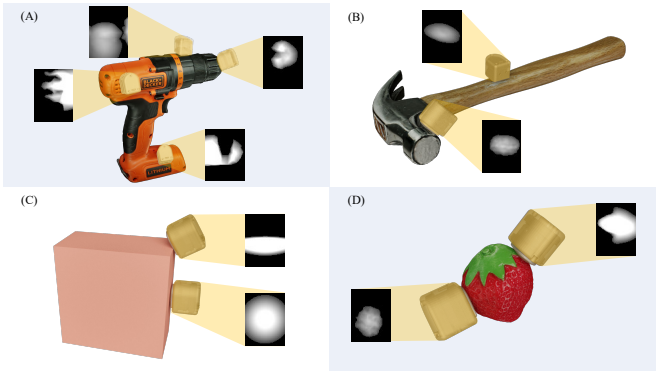


Fig. 2: Variety of textures in simulation. (C) One of the primitive objects and tactile depth readings when sensor is touching a flat surface vs an edge. (A, B, D) multiple random poses on some YCB objects and their tactile depth readings, a noticeable distribution shift becomes apparent when comparing plain primitive objects with the real textures on YCB objects. However, Tab.I indicates that AcTExplore has been generalized enough to adapt to unseen objects.

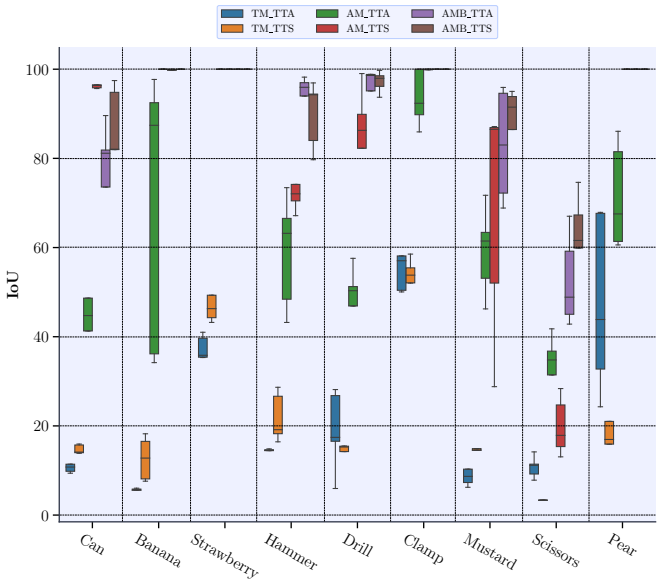


Fig. 3: **Distribution graph:** Since our proposed algorithm is not a deterministic method, we performed 5 times repeated experiments with each object. Overall, AMB methods show small variation and outperform others.

C. Further Results

1) Exploration Bonus: As we have discussed in Sec. III-D, explicitly defining $N(\mathcal{P}, a)$ as the count of times the agent took action a precisely at pose $\mathcal{P}$ throughout the trajectory history is not feasible. This is attributed to the high-dimensionality of the workspace and the likelihood that the agent might not re-encounter pose $\mathcal{P}$. As an alternative approach, we introduced $\hat{N}(\mathcal{P}, a)$, denoting the count of times the agent executed action a in proximity to pose $\mathcal{P}$. Let's define the sensor's pose as

$$\mathcal{P}_t = [T_t | R_t] \tag{2}$$

where $T_t = (x_t, y_t, z_t)$ and $R_t = (\gamma_t, \theta_t, \psi_t)$ is translation and orientation of the sensor at step t respectively. Then $\mathcal{P}_{t'}$ is a close pose to $\mathcal{P}_t$ when it satisfies the following conditions:

1) $\|(x_t - x_{t'}, y_t - y_{t'}, z_t - z_{t'})\| \leq trans_{thres}$
2) $\arccos(\min(1, \langle R_t, R_{t'} \rangle)) \leq rot_{thres}$
3) $a_t = a_{t'}$

Then we can define

$$\hat{N}(\mathcal{P}, a) = \sum_{t=0}^{T} \mathbb{I}_{close}(\mathcal{P}, \mathcal{P}_t).\mathbb{I}(a = a_t) \tag{3}$$

$trans_{thresh}$ and rot_{thresh} needs to be tuned based on sensor's sensing area and translation (T_s) and rotation (R_s) of action space (Sec. I-A). In our experiments, we used $trans_{thresh} = 2 * T_s$ and $rot_{thresh} = 4 * R_s$.

D. Metrics

a) 3D Surface IoU: We introduce 3D surface IoU metric to evaluate our method. We define a set of ground truth point clouds uniformly sampled from target object as $\mathcal{O}^{gt} = \{p_i^{gt}\}_{i=1}^{10^5}$ and $\mathcal{O}_t^s = \bigcup_{i=1}^{t} O_i = \{p_i^s\}_{i=1}^{tM}$ is the union of observed point cloud data set from initial time to time t, where $p_i^{gt}, p_i^s \in \mathbb{R}^3$ are a single point cloud data and M is the number of point clouds computed from observation O_t depth image. Then the ground truth point cloud covered set by sensor at time t is defined as

$$\mathcal{O}_t^c := \{p_i^{gt} : ||p_i^{gt} - p_i^s||_2 \leq \delta, p_i^{gt} \in \mathcal{O}^{gt} \text{ and } p_i^s \in \mathcal{O}_t^s\} \tag{4}$$

Finally, the surface IoU at time t is $\text{IoU}_t := \frac{|\mathcal{O}_t^c|}{|\mathcal{O}^{gt}|}$. Here, we used $\delta = 5$ mm.

b) Chamfer-L1 Distance: Another metric we used to evaluate our model is Chamfer-L1 distance [2]. We define the Chamfer-$L1$ distance C_t between the two 3D point cloud set $\mathcal{O}^{gt}$ and $\mathcal{O}_t^s$ at time t is defined as follows:

$$C_t := \frac{1}{2|\mathcal{O}^{gt}|} \sum_{p^{gt} \in \mathcal{O}^{gt}} \min_{p^s \in \mathcal{O}_t^s} ||p^s - p^{gt}|| + \frac{1}{2|\mathcal{O}_t^s|} \sum_{p^s \in \mathcal{O}_t^s} \min_{p^{gt} \in \mathcal{O}^{gt}} ||p^s - p^{gt}|| \tag{5}$$

II. Experiments

A. Ablation Study

We ablated the training performance of various primitives shapes on AMB-TTA model, as depicted in Fig. 1. The Cube-only model exhibited unstable IoU. Conversely, both the Cube-only and Cube + Sphere models showed early stabilization in terms of IoU. Moreover, the Cube + Sphere training model demonstrated a shorter average length, while maintaining a 90 % IoU, implying a more effective exploration of the objects within fewer steps during training which means having Cube+Sphere results in better generalization even for exploring the training objects.

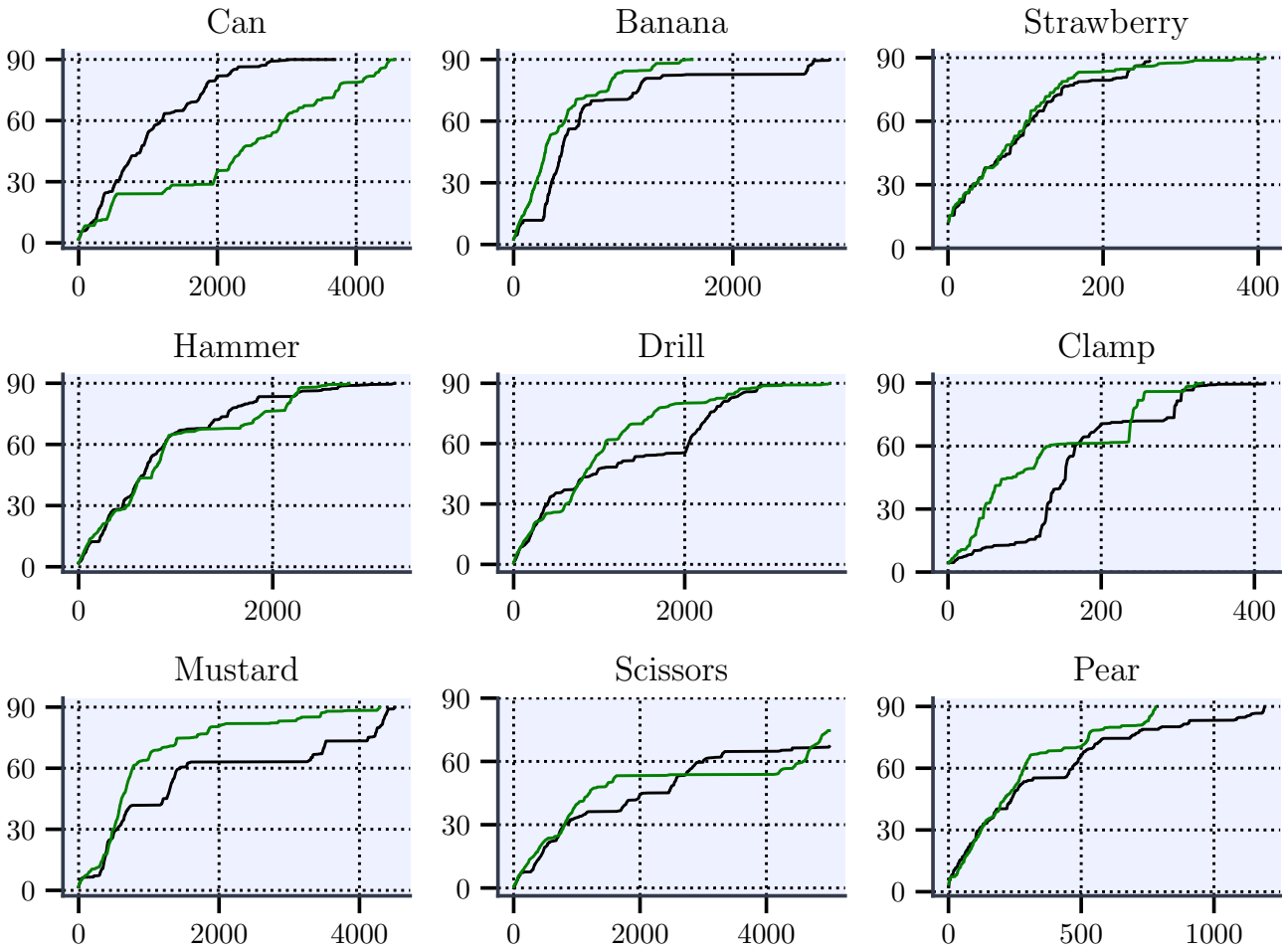


Fig. 4: **IoU-Step graph** includes AMB-TTA (in black) and AMB-TTS (in green) models, both reaching either 90 % IoU or 5,000 steps. The horizontal axis represents the number of steps, and the vertical corresponds to the IoU. Small objects, such as strawberries, achieve 90 % IoU comparably faster than large objects, such as cans.

B. Simulation Environment

We evaluated the AcTExplore on various YCB objects after training on primitive objects. Fig. 2 illustrates the diversity of shapes and textures of training and testing environments.

References

[1] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006.

[2] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

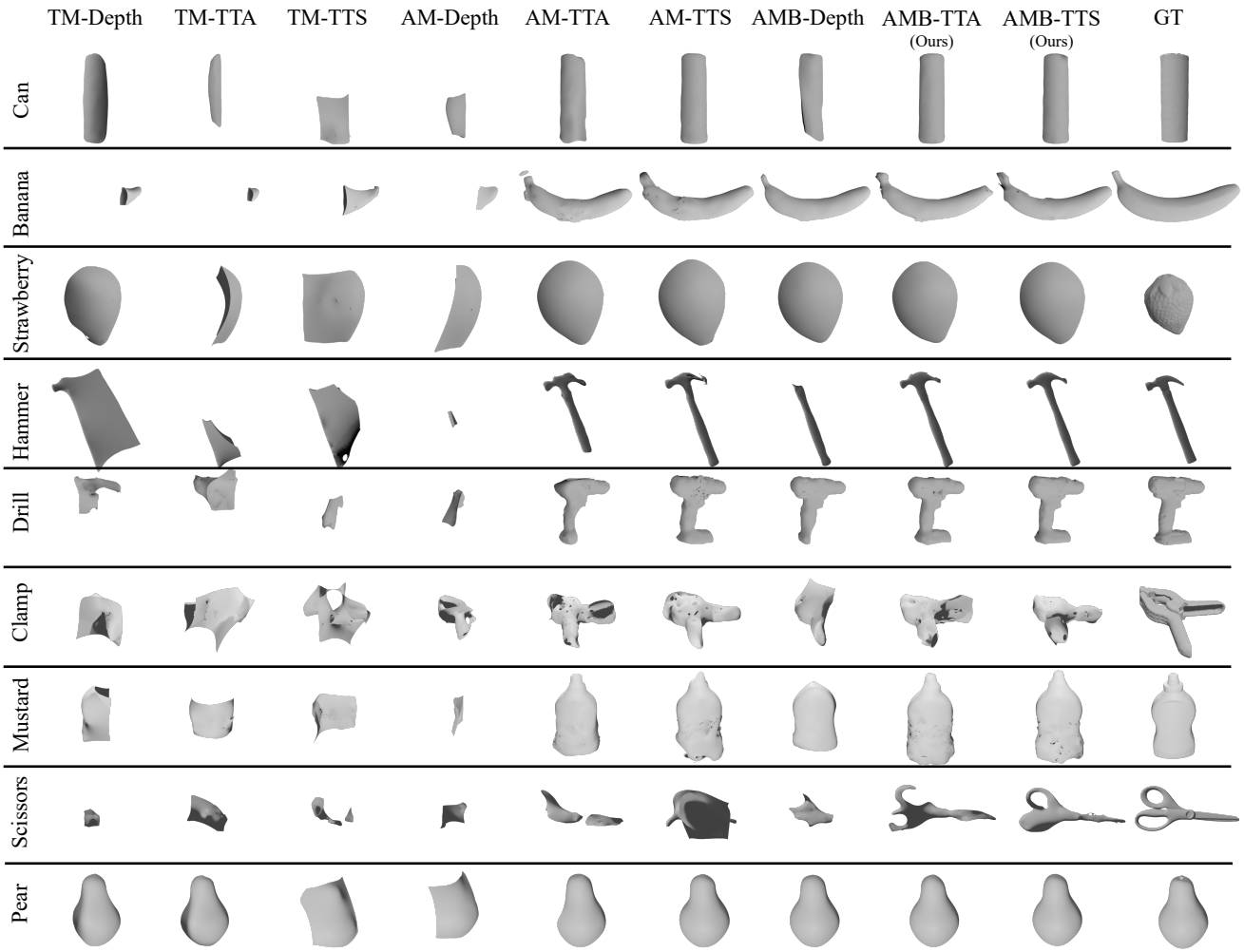


Fig. 5: **Further qualitative results on unseen YCB objects** with different state and reward settings. From active tactile exploration, we obtain point cloud data of tactile depth readings on the object's surface. To generate mesh, we apply Poisson surface reconstruction algorithm [1].

TABLE I: **Further Quantitative results on unseen YCB objects:** The table presents IoU and Chamfer-L1 distance (cm) [2] values of the predicted meshes and ground-truth meshes. The results were obtained within 5,000 steps. The surface area is listed below each object's name. Lastly, since a recurrent structure is an alternative approach to process temporal information, we implement a PPO variant with LSTM modules to compare with our proposed temporal state representations (TTA/TTS)

Models \ Objects		Can (616 cm^2)	Banana (216 cm^2)	Strawberry (68 cm^2)	Hammer (410 cm^2)	Drill (591 cm^2)	Clamp (111.13 cm^2)	Mustard (454.54 cm^2)	Scissors (165.48 cm^2)	Pear (172.47cm^2)
		IoU ↑ (Chamfer-L_1 ↓)								
TM	depth	31.93 (2.66)	11.11 (7.52)	83.60 (0.44)	32.78 (1.86)	19.19 (4.1)	81.76 (1.5)	10.07 (4.07)	24.29 (8.15)	70.95 (0.55)
	TTA	17.60 (3.57)	6.03 (9.03)	41.0 (1.23)	14.85 (6.94)	28.15 (3.99)	86.29 (1.4)	19.94 (3.22)	14.17 (4.98)	67.89 (0.62)
	TTS	15.93 (5.22)	18.23 (5.48)	57.89 (0.88)	28.66 (2.47)	15.5 (3.97)	58.53 (1.08)	14.55(2.95)	11.26(4.97)	30.13(2.13)
AM	depth	11.59 (5.49)	10.22 (6.84)	47.33 (1.16)	5.07 (7.69)	9.49 (4.03)	58.81(0.82)	11.04(5.16)	5.11(6.78)	28.75(2.08)
	TTA	72.70 (0.56)	97.70 (0.35)	**100** (0.28)	79.80 (0.82)	57.58 (1.43)	**100** (0.62)	71.72 (0.80)	41.77 (2.87)	86.07(0.43)
	TTS	**98.25** (0.22)	**100** (0.34)	**100** (0.31)	88.22 (0.44)	99.02 (0.37)	**100** (0.69)	87.13 (0.59)	28.37 (2.38)	**100** (0.23)
AMB	depth	41.45 (1.42)	98.64 (**0.25**)	**100 (0.23)**	61.42 (1.17)	79.68 (0.95)	44.76 (1.77)	65.74 (0.9)	31.99 (3.2)	**100** (**0.2**)
	depth+LSTM	88.54 (0.3)	99.96 (0.28)	**100** (0.24)	87.54 (0.49)	92.81 (0.36)	99.55 (**0.56**)	88.33 (0.36)	29.83 (0.58)	**100** (**0.2**)
	TTA (ours)	89.6 (0.29)	**100** (0.33)	**100** (0.25)	**98.22** (0.29)	98.85 (0.32)	**100** (0.66)	**95.91** (0.51)	67.02 (0.87)	**100** (0.22)
	TTS (ours)	97.45 (**0.20**)	**100** (0.3)	**100** (0.25)	96.96 (**0.28**)	**99.74 (0.31)**	**100** (0.59)	95.02 (**0.49**)	**74.62** (**0.61**)	**100** (**0.2**)