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.

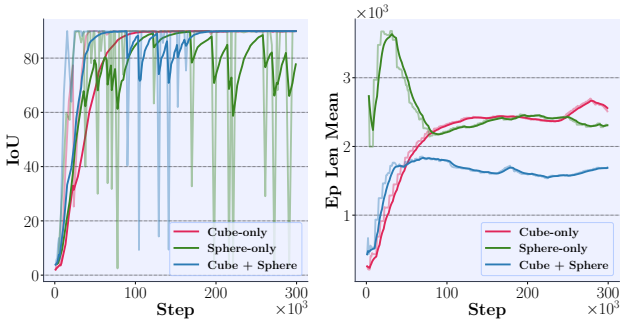

Fig. 1: **Ablation study of training primitives:** We trained AMB-TTS with cube-only and sphere-only setting as well.

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.

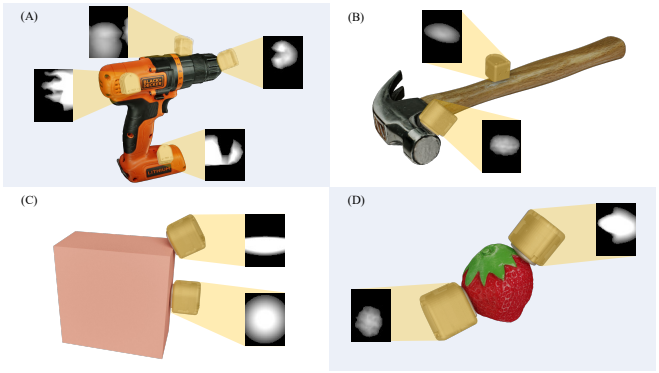

Fig. 2: Variety of textures in simulation. (C) One of the primitive objects and tactile depth readings when sensor is touching a flat surface vs an edge. (A, B, D) multiple random poses on some YCB objects and their tactile depth readings, a noticeable distribution shift becomes apparent when comparing 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.

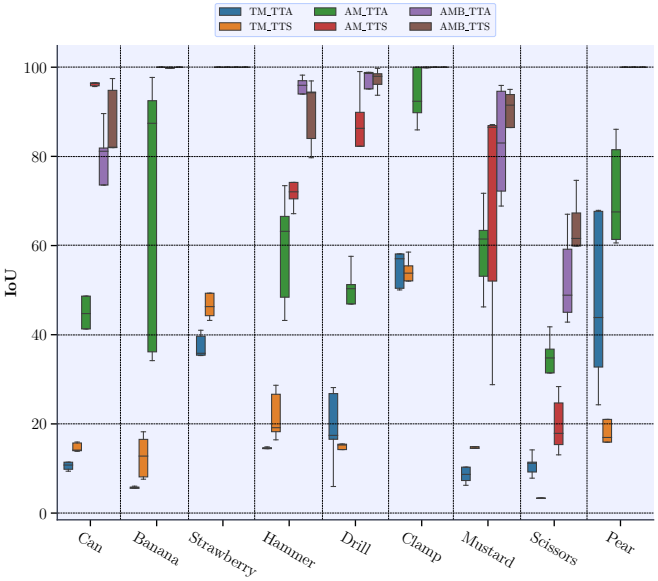

Fig. 3: **Distribution graph:** Since our proposed algorithm is not a deterministic method, we performed 5 trials with each object. Overall, AMB reward function shows small variation and outperform others.

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

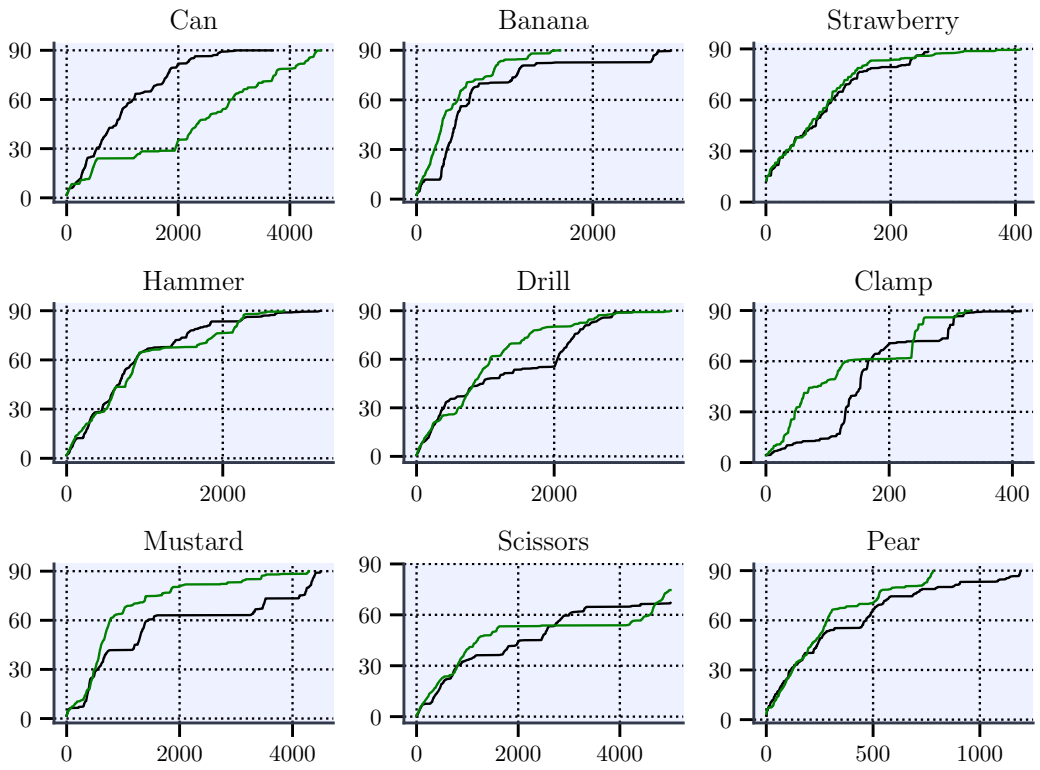

Fig. 4: **IoU-Step graph** includes AMB-TTA (in black) and AMB-TTS (in green) models, both reaching either 90 % IoU or 5,000 steps. The horizontal axis represents the number of steps, and the vertical corresponds to the IoU. Small objects like strawberry, achieve 90 % IoU comparably faster than large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

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.

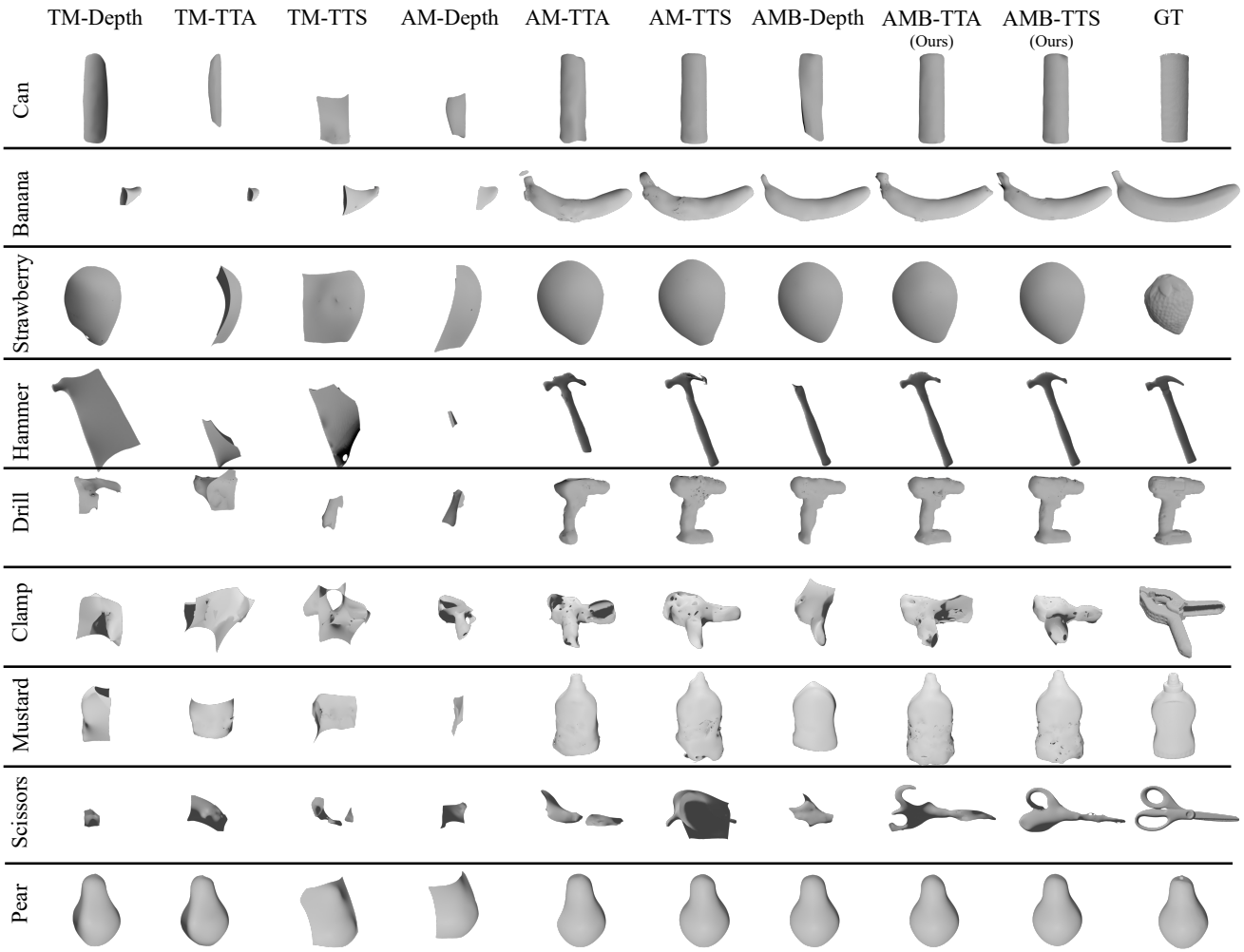

Fig. 5: **Further qualitative results on unseen YCB objects** with different state and reward settings. From active tactile exploration, we obtain point cloud data of tactile depth readings on the object's surface. To generate mesh, we apply Poisson surface reconstruction algorithm [1].

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**)