

Supplementary Materials

Anonymous Author(s)

Affiliation

Address

email

Contents

1	Tactile Sensor Hardware	1
1.1	Tactile Sensor Manufactory	1
1.1.1	Tactile Sensor Pad Design	1
1.1.2	Reading Board Design	2
1.2	Tactile Hardware Evaluation Experiment	2
1.2.1	Physical Characteristics	2
1.2.2	6-DoF Object Pose Estimation	3
2	Experiment details for Imitation Learning	4
2.1	System Overview	4
2.2	Experiment Setup Details	5
2.2.1	Details for the Egg Steaming task	5
2.2.2	Details for the Fruit Preparation Task	6
2.2.3	Details for the Hex Key Collection Task	7
2.2.4	Details for the Sandwich Serving Task	9
2.3	Learning Details	9

1 Tactile Sensor Hardware

1.1 Tactile Sensor Manufactory

1.1.1 Tactile Sensor Pad Design

The tactile sensing pads leverage a triple-layer design, where a piezoresistive layer (Velostat) is sandwiched between two sets of orthogonally aligned conductive yarns serving as electrodes. During the tactile sensor manufacturing, we first align 16 Stainless Thin Conductive Threads on top of the Velostat layer and then use high-strength adhesive (3M 468MP) to ensure robust electrical contact between the electrodes and the Velostat layer. Additionally, we use adhesive to secure the conductive thread connections to the connector. The connector links all the threads to a flexible flat cable, allowing the signal to be transmitted to the PCB board. This design makes the wires of our tactile sensor highly flexible, facilitating easier installation in various locations, such as the robot end-effector, which requires constant movement during manipulation. To ensure the tactile sensor’s long-term robustness, we attach a polyimide layer on top of the adhesive. Polyimides are known for their thermal stability, good chemical resistance, excellent mechanical properties, and characteristic

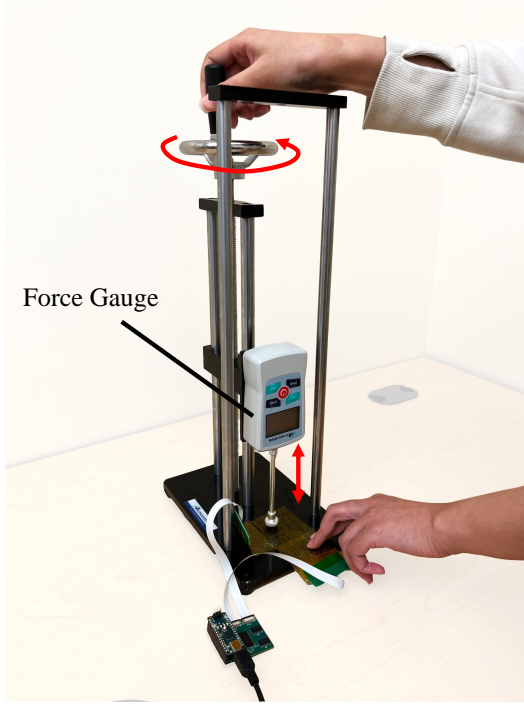


Figure 1: Tactile Physical Characteristics Evaluation Experiment

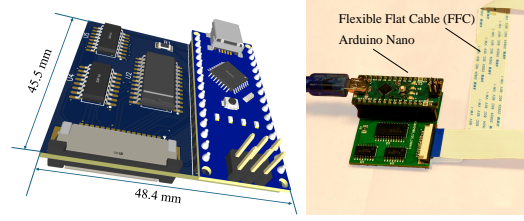


Figure 2: Tactile Reading Board Design.

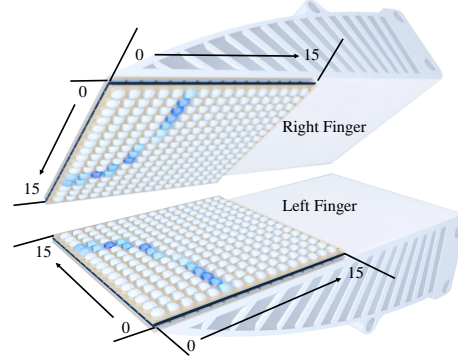


Figure 3: Tactile Sensor Order Visualization.

31 orange/yellow color. After completing these steps, we finish aligning the 16 threads for the rows.
 32 Then, we flip the sensor and align the 16 threads for the columns.

33 After obtaining the tactile sensor pad, we attach the sensors to the robot fingers. The order of each
 34 tactile sensor unit is visualized in Fig. 3. We clearly define the tactile order to ensure that each
 35 sensor's position can be accurately calculated, and the tactile signals can correctly correspond to our
 36 real setting and dataset.

37 1.1.2 Reading Board Design

38 To ensure easy installation of the tactile reading board in the robot, we have designed it to be as
 39 compact as possible, as shown in Fig. 2. The tactile reading board measures 45.5 mm \times 48.4 mm
 40 and includes an Arduino. The small size further enhances the scalability of our tactile sensors. We
 41 use two 8-bit shift registers and one 16-channel analog switch to process the tactile signals, which
 42 are then input to the Arduino. The ADC in the Arduino converts the analog signals from the tactile
 43 sensor into digital signals and forwards them to the host via serial communication. We will release a
 44 comprehensive reading board scheme so that the community can directly order from a PCB supplier
 45 to easily replicate our tactile sensor.

46 1.2 Tactile Hardware Evaluation Experiment

47 1.2.1 Physical Characteristics

48 To investigate the physical characteristics of our tactile sensors, we designed two experiments. As
 49 illustrated in Fig. 1, we use a force gauge to apply specific force on the tactile sensor surface. The first
 50 experiment tests how individual tactile sensor units react to applied force. The second experiment
 51 aims to test the consistency of the entire sensor pad, showcasing the variance between different
 52 regions on the tactile sensor pad.

53 **Individual Sensor Performance.** We began by randomly selecting 10 sensors from a total of 256
 54 sensors in one sensor pad. For each selected sensor, we applied a normal force incrementally, ranging

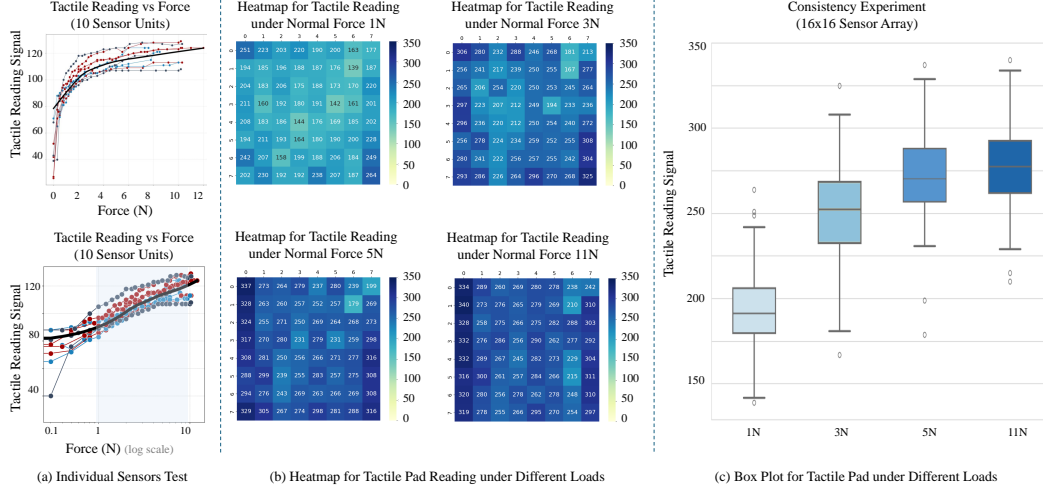


Figure 4: **Results of Physical Characteristics Experiments.** Part (a) shows the results of individual sensors’ performance according to the force applied to their surface. Part (b) demonstrates the tactile sensor pad’s consistency under different normal forces. Each heatmap displays the tactile sensor pad’s readings in an 8×8 grid, where each number represents the sum of four sensor units. Part (c) presents the results from part (b) in a single figure, illustrating the mean and standard deviation.

from 0 to 12 N, and recorded the stable tactile reading accordingly. Each sensor generated an average of 24 data points. This method allowed us to observe the individual sensor’s response to varying force levels and identify its saturation thresholds. As shown in Fig. 4 (a), we plot tactile reading versus normal force and identified that the saturation zone begins when the normal force exceeds 9 N. The fitting curve for the 10 sensors is depicted in the black line. Additionally, we applied a logarithmic scale to the x-axis (normal force), resulting in an approximately linear region for normal forces from 1 N to 9 N. The region is highlighted with a blue background, as illustrated in Fig. 4 (a).

Tactile Sensor Pad Consistency. In the second part of the experiment, we used a 16×16 tactile sensor pad and divided it into 8×8 blocks, with each block comprising 4 sensors (2×2 matrix area). Uniform loading was applied across each block using the force gauge with a circular contacting area of 176.7 mm^2 . For each 2×2 block, we collected the sum of the four tactile readings from individual sensors, enabling us to generate a heat map that visualizes the sensor response under specific loading conditions across the entire pad. Four different loading conditions (1 N, 3 N, 5 N, and 11 N) were applied to comprehensively assess the overall performance, providing a detailed representation of the resolution under varying forces. For each force condition, we measured once for each block, resulting in a total of 64 data points per condition. We then generate a heatmap for each force condition as shown in Fig. 4 (b). We calculated the mean and standard deviation for these data points and removed outliers. Finally, as illustrated in Fig. 4 (c), we generated a box plot from 4 sets of 64 tactile readings, demonstrating its consistency across the entire sensor and the stable functionality of the tactile sensors.

1.2.2 6-DoF Object Pose Estimation

In the main paper, we demonstrated the effectiveness of dense, continuous tactile information for fine-grained manipulation tasks. To gain a more comprehensive understanding of the information captured by our proposed sensors, we conducted additional experiments on 6-DoF object pose estimation. These experiments revealed that the sensors embed information about object geometry and local contact patterns, which is crucial for manipulation tasks requiring robust and adaptive grasping as well as precise in-hand reorientation behavior.

Specifically, we define the task as estimating the 6-DoF pose of an object using only tactile observations, **without** any visual input. We assume that the object geometry is known and denote its 3D point cloud as $P^{\text{obj}} \in \mathbb{R}^{N \times 3}$. The tactile observation, obtained by filtering the tactile-based point cloud according to the activation value, is denoted as $P^{\text{tactile}} \in \mathbb{R}^{M \times 3}$. Our objective is to track the

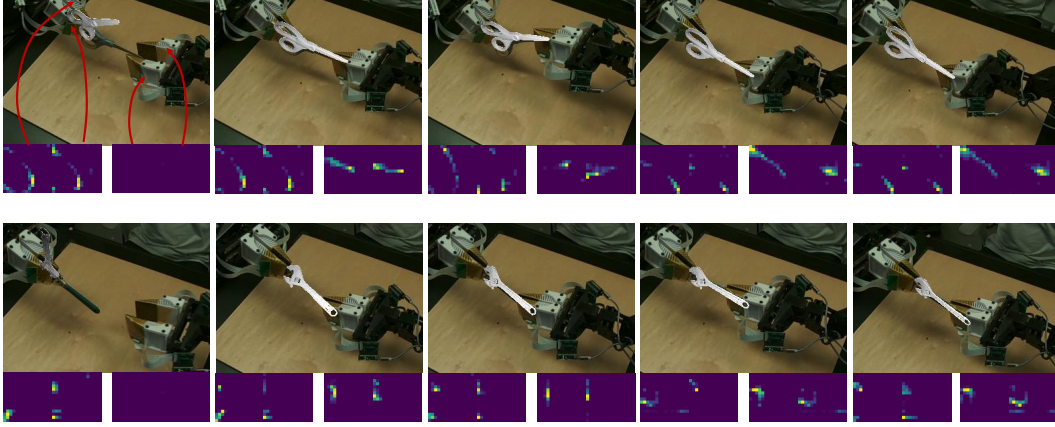


Figure 5: **Pose Estimation.** In this experiment, we estimate the object pose **without** vision information. We can see that our pose estimation becomes more accurate as we have more complete tactile signals. We can also track the object’s pose as it rotates. Through this estimation, we demonstrate that our hardware can be potentially used for in-hand pose estimation and other visuotactile tasks.

86 pose of the object in the 3D space, $\mathbf{T} \in \mathbb{SE}(3)$, where,

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \in \mathbb{SE}(3), \quad (1)$$

87 in which the Euclidean group $\mathbb{SE}(3) := \{\mathbf{R}, \mathbf{t} \mid \mathbf{R} \in \mathbb{SO}^3, \mathbf{t} \in \mathbb{R}^3\}$.

88 We solve the pose-tracking problem using particle filtering [1]. We first define our observation
89 function $P^{\text{obs}} = f(\mathbf{T})$ and then the weighting functions $w = g(P^{\text{obs}}, P^{\text{tactile}})$ as follows:

$$\begin{aligned} f(\mathbf{T}) &= \mathbf{R}P^{\text{obj}} + \mathbf{t}, \\ g(P^{\text{obs}}, P^{\text{tactile}}) &= \sum_{p_i \in P^{\text{tactile}}} \min_{p_j \in P^{\text{obs}}} \|p_i - p_j\|^2. \end{aligned} \quad (2)$$

90 The observation function transforms the object model point cloud using \mathbf{T} , while the weighting
91 function calculates the distance from the contact points to the observation points. In practice, we
92 scale the weights using an exponential function to facilitate convergence. Given the observation and
93 weighting functions, we employ a standard particle filter to determine the object’s pose.

94 Some example results are shown in Figure 5. Before the right-side robot makes contact with the
95 object, we can only rely on the tactile signals from the left-side robot. Therefore, there are a lot of
96 plausible solutions. Although our estimated pose is one of the plausible solutions given the one-side
97 tactile signal, the estimation is still inaccurate. When the right-side robot contacts the object, our
98 estimated pose aligns well with visual observation. Also, when the object is rotating in the hand, the
99 object pose is tracked accurately.

100 2 Experiment details for Imitation Learning

101 2.1 System Overview

102 As shown in Fig. 6, We employ a bimanual teleoperation system with three Realsense cameras
103 and four tactile sensor pads on four robot fingers. Our tactile signal communication is facilitated
104 by a multi-threaded ROS (Robot Operating System) node. This node captures tactile signals and
105 publishes them at a frequency of 30 Hz. All data, including that from cameras and tactile sensors, is
106 collected through multi-threading. Each data frame received is timestamped, and after an episode is
107 completed, we align all data with these timestamps. This synchronization is crucial for maintaining
108 the consistency of the multimodal dataset, enabling accurate temporal alignment between tactile
109 feedback and visual data. To manage the heavy load of processing frames from three cameras, we



Figure 6: **System Overview.** We attach four tactile sensors to four robot fingers and install three Realsense cameras to cover the workspace. All the objects used for the task are shown in the workspace. Additionally, we install a background screen to display the tactile feedback.

110 collect data at 10 Hz to ensure consistency. We set a top camera (Realsense 455) to cover the entire
 111 workspace and positioned two other cameras (Realsense 435) close to the workspace to capture more
 112 detailed information. When using point cloud data from multiple cameras, we incorporate data from
 113 all cameras. For the baseline method using a single camera, we use only the top camera.

114 We also implement real-time tactile information feedback, as shown in Fig. 6 (a). During data
 115 collection, tactile signals are visually displayed on the operator’s screen, enabling them to assess the
 116 adequacy of contact for secure grasping. Additionally, during the policy rollout, this visualization
 117 helps us see in real-time how tactile information relates to robot motion.

118 2.2 Experiment Setup Details

119 In this section, we discuss the detailed information of the four tasks described in the main paper.
 120 Each task consists of four steps, as illustrated in Fig 8. We will discuss the motions and evaluation
 121 metrics for each step, and highlight how these steps demonstrate the capabilities of our tactile sensors.
 122 The typical failure cases are shown in Fig. 7 and will be discussed in the following sections.

123 2.2.1 Details for the Egg Steaming task

124 *Step 1: Open Egg Tray.* The robot uses its right hand to open the egg tray, which mirrors the common
 125 scenario where the egg is often occluded by the tray. This realistic setup is maintained to reflect daily
 126 life, avoiding task simplification. *Evaluation Metrics:* The robot must open the tray sufficiently to
 127 allow its fingers to grasp the egg. Failure to open the tray adequately will result in the subsequent
 128 task failing. The initial position of the egg tray will be randomized within an area of 7-10 cm during
 129 both data collection and policy rollout.

130 *Step 2: Grasp Egg.* The robot uses its right hand to grasp the egg in the tray. This motion is complex,
 131 requiring the robot to slowly increase the force and carefully grasp the egg despite heavy occlusion.
 132 The robot with a visuo-tactile policy will retry if there is no stable tactile signal in hand, while
 133 a vision-only policy may proceed to the next goal due to heavy occlusion as shown in Fig. 7 (a).
 134 *Evaluation Metrics:* The robot can reattempt to grasp the egg, but the step fails if it moves to the next
 135 stage without the egg or if the egg falls during the transition from the tray to the steaming machine.
 136 Additionally, prolonged time spent in the egg tray will also be considered a failure.

137 *Step 3: Place Egg.* The robot needs to safely place the egg in the steaming machine, which already
 138 contains two eggs. It must avoid causing the other eggs to fall while placing the egg in-hand. This
 139 step highlights our flexible thin sensor’s capability to perform fine-grained tasks in narrow spaces. As
 140 the robot hand exits the steaming machine, tactile information ensures there is no contact between
 141 the egg and the gripper, signaling the robot to proceed to the next stage. In contrast, a vision-only
 142 policy may cause confusion about whether the robot can move out safely, potentially prolonging its
 143 stay in the steamer and increasing the risk of dislodging the other eggs. *Evaluation Metrics:* The
 144 robot can place the egg anywhere inside the steaming machine, but the step fails if the robot does not
 145 place the egg in the steaming machine or if it causes the other eggs to fall to the ground.

146 *Step 4: Cover the Steaming Machine.* The robot needs to use its left hand to grasp the cover of the
 147 steaming machine and place it safely inside. This task is challenging due to the unique shape of the
 148 steaming machine’s handle, as shown in Fig. 7 (a). The robot must apply a precise amount of force to
 149 the handle: sufficient to lift it but not so much that the cover flips and falls to the ground. The robot
 150 must apply a precise amount of force to the handle: sufficient to lift it but not so much that the cover
 151 flips and falls to the ground. This step showcases how our tactile sensor enables the robot to perform
 152 fine-grained grasping manipulations, similar to a human’s ability to apply suitable and stable force to
 153 grasp objects. *Evaluation Metrics:* The robot is allowed multiple attempts to grasp the cover. The
 154 task is considered successful if the cover is securely placed on the steaming machine. It is considered
 155 a failure if the cover flips or falls during the process.

156 2.2.2 Details for the Fruit Preparation Task

157 *Step 1: Grasp and Place the Plate.* The robot needs to use its left hand to grasp and place the plate
 158 on the table. This step introduces additional randomization and variance due to the varying positions
 159 of the plates, increasing the task’s complexity. *Evaluation Metrics:* The task is considered successful
 160 if the robot grasps the plate and places it on the table.

161 *Step 2: Open Plastic Bag.* The robot needs to use its two hands to cooperate together to open the
 162 bag. The plastic bag is transparent and usually adds additional noise to the point cloud. *Evaluation*
 163 *Metrics:* The task is considered successful if the robot opens the bag wide enough for the gripper to
 164 get in.

165 *Step 3: Grasp the fruit.* The robot needs to use its right hand to get inside the plastic bag and grasp
 166 the fruit. This step is the most important and difficult in this task. First, as shown in Fig. 8(Task
 167 2: Fruit Preparation), the robot and manipulated objects are highly occluded in the bag, making
 168 it impossible for visual information to observe critical details. Our visuo-tactile policy will grasp
 169 multiple times until there is stable tactile information to secure the grapes, while a vision-only policy
 170 typically attempts the motion once regardless of the presence of grapes, making the grasping success a
 171 random event. Second, the grapes are usually clustered together, requiring the robot to apply suitable
 172 force to avoid damaging the fruit. Our visuo-tactile policy can successfully grasp single or multiple
 173 grapes from the bag, while a vision-only policy may break the grapes when the robot grasps multiple
 174 grapes (as shown in Fig. 7 (b)). since it aligns the gripper joint states instead of using force-related
 175 information. Third, this task also showcases our sensors’ human-like dexterous manipulation; our
 176 tactile-integrated gripper is thin enough to get into the gaps between grapes, making it easier to
 177 grasp the grapes in a cluster. *Evaluation Metrics:* The task is considered successful when the robot
 178 successfully grasps the fruit out of the plastic bag. The task is considered a failure if the robot breaks
 179 the grapes or moves to the next stage without the grapes. The policy also fails if the robot stays in

Typical Failure Cases of Baselines

(We visualize tactile signals in the screen no matter if tactile modality in the observation)



Figure 7: **Failure Cases.** We present typical failure cases of the baseline method for all four tasks and analyze the reasons for these failures to highlight the complexity of the tasks and the importance of tactile feedback during these steps.

180 the bag for a long time without moving, which usually happens with the vision-only policy that is
181 confused about the states of the objects and the robot end-effector under high occlusion.

182 *Step 4: Place grape.* The robot needs to place the grapes on the plate. The task may fail if the robot
183 uses too much force to grasp the grapes, causing them to stick in the gripper and resulting in failure.

184 *Evaluation Metrics:* The task is considered successful if the robot successfully places the grapes on
185 the plate and returns to the initial position.

186 2.2.3 Details for the Hex Key Collection Task

187 *Step 1: Right Hand Grasp.* The robot needs to use its right hand to grasp the tail of the hex key
188 and lift it stably to the middle of the air. The initial position of the hex key is tricky, but it reflects
189 a common daily life scenario where only the tail of the hex key is accessible, requiring additional
190 adjustments to insert the hex key properly. A typical failure case of baselines, shown in Fig. 7 (c),
191 occurs when the robot does not secure a stable grasp, resulting in significant slippage during the
192 lifting process. Even if the hex key remains in-hand, this slippage can cause subsequent task failures.
193 One observation during the experiment is the consistent small slippage during the first grasp, leading
194 to variations in the hex key’s in-hand pose, which adds complexity to the following steps. *Evaluation*
195 *Metrics:* The robot successfully grasps the hex key without significant slippage.

196 *Step 2: Left Hand Grasp.* The robot needs to use its left hand to grasp the head of the hex key to
197 ensure the following adjustment step. *Evaluation Metrics:* The robot left hand successfully grasp the
198 hex key.

199 *Step 3: In-hand Adjustment.* The robot’s left and right hands need to cooperate to adjust the hex key’s
200 position so that it is in a ready pose for the following insertion. Our goal is to adjust the hex key to be
201 perpendicular to the robot’s fingers, making the subsequent insertion task easier. The vision-only
202 policy usually fails to adjust the position correctly, shown in Fig 7(c), making the following insertion
203 impossible. *Evaluation Metrics:* The robot’s two arms must cooperate to adjust the hex key’s pose.
204 The final pose should have a sufficiently long tail, and the hex key should be almost perpendicular to
205 the robot’s fingers.

206 *Step 4: Insertion.* This step is complex because the pose of the hex key in hand varies, even if the
207 robot successfully adjusts the hex key’s position in the last step. A successful policy can implicitly
208 reference the hex key’s position in hand and make the necessary adjustments for insertion. *Evaluation*
209 *Metrics:* The robot successfully inserts the hex key into the hole rather than placing it on the table or
210 getting stuck during the insertion process.

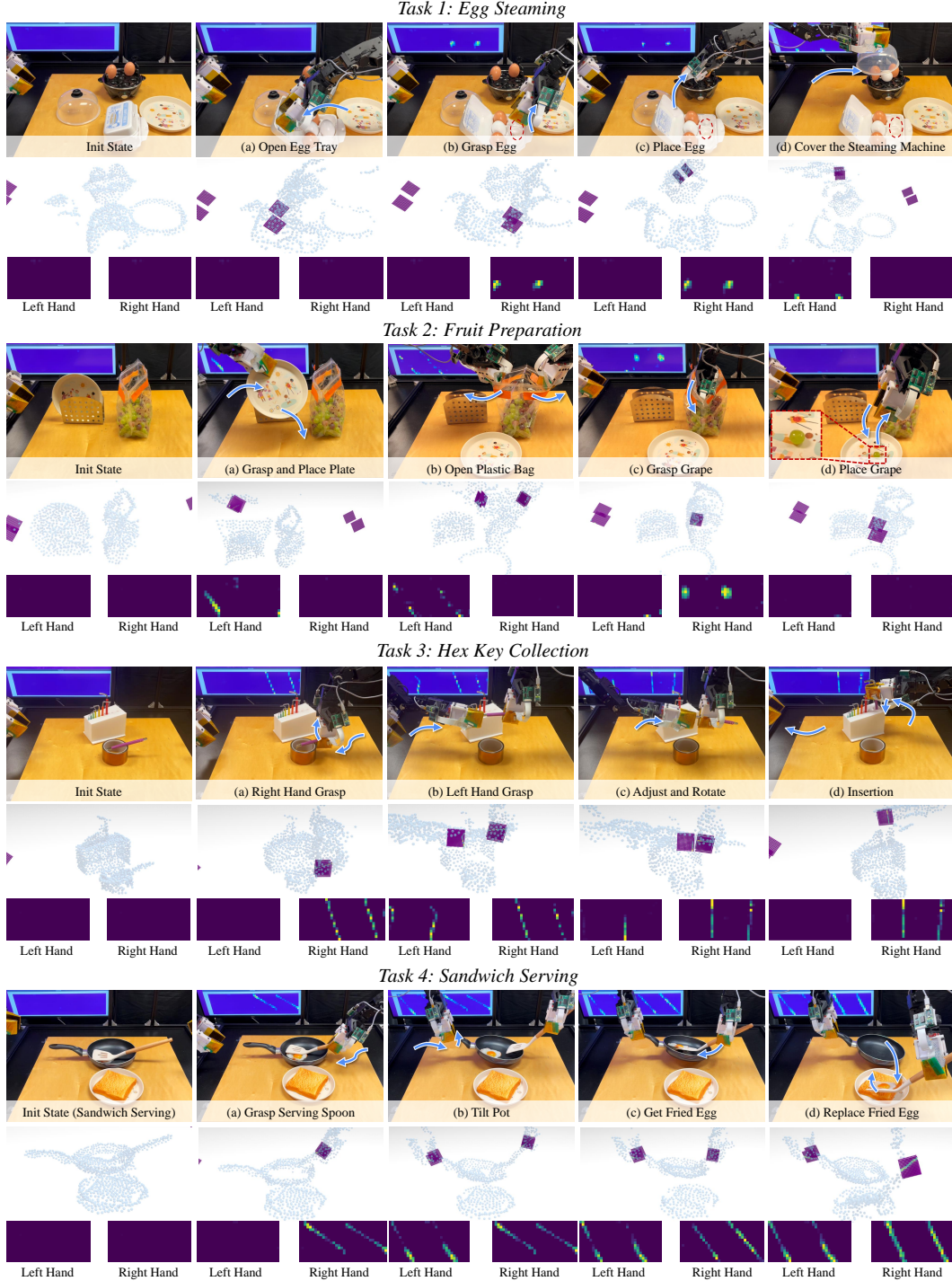


Figure 8: Quantative Result of Tactile Representation. Here we showcase a total of four tasks. For each task, the first row presents the real image. In the second row, we visualize our visuo-tactile points in a unified 3D space to demonstrate how tactile points can infer spatial relationships between objects and contact areas. The third row provides a 2D image to clearly visualize the tactile signals.

2.2.4 Details for the Sandwich Serving Task

Step 1: Grasp Serving Spoon. The robot needs to use its right hand to grasp the spoon and lift it into the air. *Evaluation Metrics:* The spoon is successfully lifted into the air with minimal slippage.

Step 2: Tilt Pot. In order to successfully obtain the egg in the next step, the robot’s left hand needs to grasp the pot’s handle and tilt the pot. The gripper should not exert excessive force to ensure that the handle does not rotate within the robot’s hand. *Evaluation Metrics:* The robot’s left hand must successfully grasp the handle and then tilt it to a certain angle.

Step 3: Get Fried Egg. The robot’s two hands need to cooperate to retrieve the fried egg. The right hand will use the spoon to reach the bottom of the pot and maneuver beneath the egg. During this process, the spoon will passively rotate in the hand. Our visuo-tactile policy can explicitly track the states of the spoon, while the baseline policy often fails due to the spoon’s rotation in the hand. *Evaluation Metrics:* The robot successfully retrieves the fried egg with the spoon.

Step 4: Replace Fried Egg. The robot needs to move the spoon to the top of the bread and tilt it to place the egg on the bread. A typical failure occurs when the robot does not perform a successful tilt motion due to changes in the spoon’s position within the hand. Our visuo-tactile policy can account for these changes and adjust the motion accordingly. *Evaluation Metrics:* The robot successfully places the fried egg on top of the bread.

2.3 Learning Details

We use Pointnet++ as the learning backbone. we employ hierarchical feature extraction and processing for point cloud data. We use three set abstraction layers: the first set abstraction layer processes 64 points with a 0.04 radius and 16 samples using a multi-layer perceptron (MLP) with layers [64, 64, 128]; the second set abstraction layer processes 16 points with a 0.08 radius and 32 samples using an MLP with layers [128, 128, 256]; the third set abstraction layer serves as a global abstraction layer with an MLP of [256, 512, 1024]. For further feature processing, we use fully connected layers: the first fully connected layer transforms 1024 features to 512, and the second fully connected layer reduces 512 features to 256. We disable batch normalization layers.

References

- [1] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.