# 3D-ViTac: Learning Fine-Grained Manipulation with Visuo-Tactile Sensing

**Binghao Huang**[1] **Yixuan Wang**[1] **Xinyi Yang**[2] **Yiyue Luo**[3] **Yunzhu Li**[1]

Columbia Univerisity[1] University of Illinois Urbana-Champaign[2]
University of Washington[3]

## Abstract

Tactile and visual perception are both crucial for humans to perform fine-grained interactions with their environment. Developing similar multi-modal sensing capabilities for robots can significantly enhance and expand their manipulation skills. This paper introduces **3D-ViTac**, a multi-modal sensing and learning system designed for dexterous bimanual manipulation. Our system features tactile sensors equipped with dense sensing units, each covering an area of $3mm^2$. These sensors are low-cost and flexible, providing detailed and extensive coverage of physical contacts, effectively complementing visual information. To integrate tactile and visual data, we fuse them into a unified 3D representation space that preserves their 3D structures and spatial relationships. The multi-modal representation can then be coupled with diffusion policies for imitation learning. Through concrete hardware experiments, we demonstrate that even low-cost robots can perform precise manipulations and significantly outperform vision-only policies, particularly in safe interactions with fragile items and executing long-horizon tasks involving in-hand manipulation. Our project page is available at https://binghao-huang.github.io/3D-ViTac/.

**Keywords:** Contact-Rich Manipulation, Multi-Modal Perception, Tactile Sensing, Imitation Learning

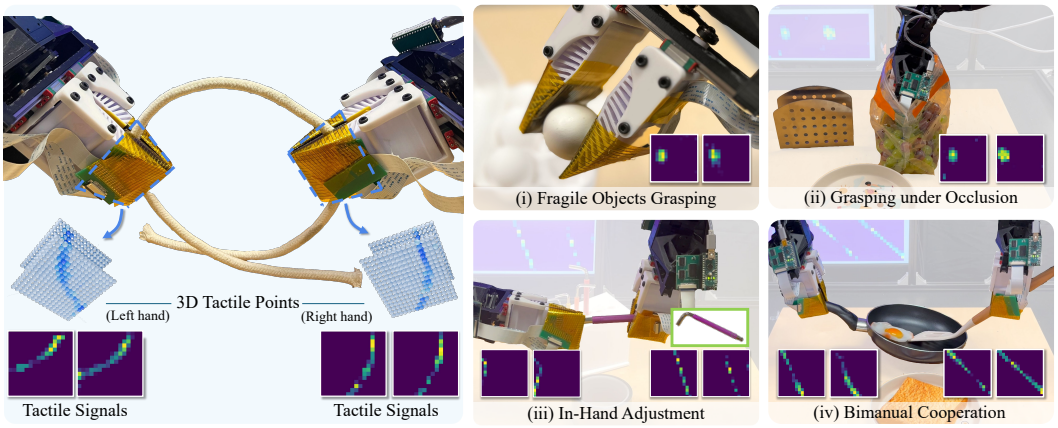
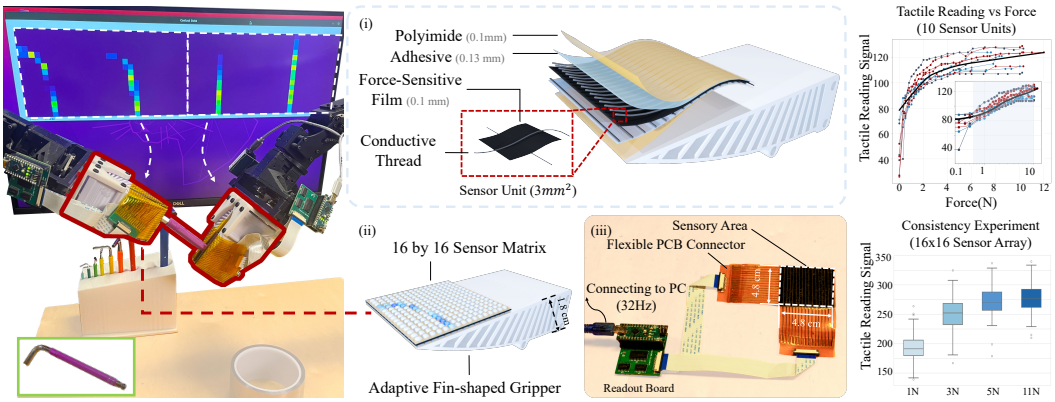



Figure 1: We propose **3D-ViTac**, a multi-modal sensing and learning system for dexterous bimanual manipulation. This system features flexible, scalable, low-cost tactile sensors, each finger equipped with a $16 \times 16$ sensor array. To demonstrate the capabilities of our system in performing precise manipulations, we showcase four tasks that utilize the force-related and in-hand position information provided by the tactile sensors.

## 1 Introduction

Humans heavily rely on both visual and tactile sensing to perform everyday manipulation tasks. Consider grasping an egg or a grape: we start by visually locating the object, and then extract more

NeurIPS 2024 Workshop on Touch Processing: a new Sensing Modality for AI .




(a) Tactile Sensing Platform (b) Tactile Integrated Soft Gripper Design (c) Physical Characteristic

Figure 2: **Our Tactile Sensing Platform. Part (a)** shows our bimanual tactile integrated system setup. We deploy four tactile sensor pads (two for each hand) on the soft grippers. The tactile readings are displayed on the back screen. **Part (b)** describes the design of our tactile-integrated soft gripper. Each sensor comprises 256 sensing units, with their locations on the gripper shown in (ii). We have also designed a readout board to collect tactile signals and forward them to the host computer. **Part (c)** shows the physical characteristics and sensing consistency of our tactile sensors (details in Sec. 4).

information from tactile interactions with the object to determine the appropriate amount of force to apply. When eating with a spoon, our eyes estimate the global position and geometric information of the spoon, while our sense of touch provides detailed contact information during interactions with the food. Vision and touch complement each other, enhancing our interaction with the environment and significantly increasing our flexibility and robustness, especially in tasks involving large occlusions and in-hand manipulation.

We conducted comprehensive evaluations on four challenging real-world tasks (e.g., manipulating fragile objects like eggs and fruit, and in-hand manipulation of tools and utensils, as shown in Fig. 1). The results demonstrate that our 3D visuo-tactile representation significantly enhances the performance of contact-rich manipulations by providing more detailed contact states and local geometry or position information. We observed that tactile information is especially critical when there is heavy visual occlusion. We also conducted detailed ablation studies regarding the sensing characteristics, comparing performance across different tactile resolutions, and showed the importance of continuous tactile reading. Additionally, the inclusion of tactile feedback during the data collection process enables the operator to gather higher-quality data, making the final policy more robust.

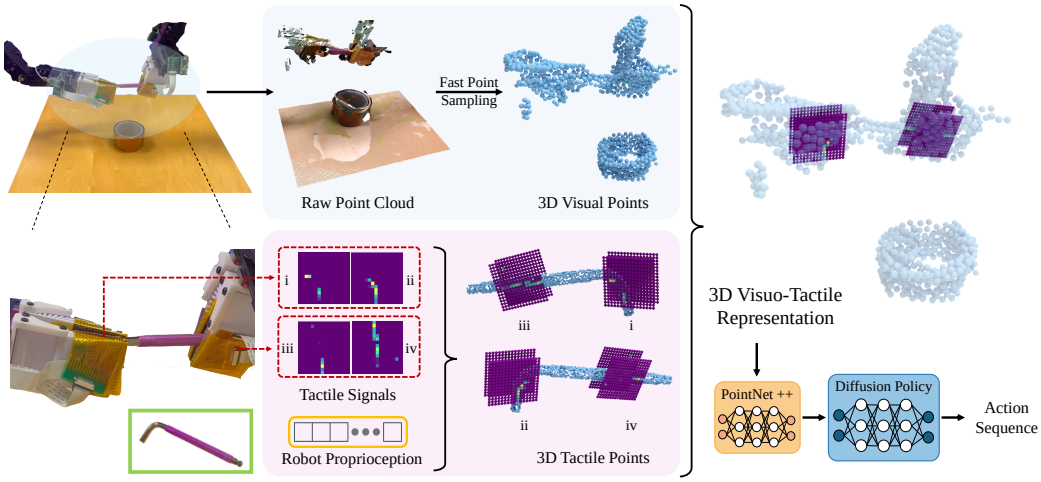
## 2 Visuo-Tactile Manipulation System

### 2.1 Sensor and Gripper Design

**Flexible Tactile Sensors.** Our sensor pads consist of resistive sensing matrices that convert mechanical pressure into electrical signals. Designed with a total thickness of less than $1mm$, the tactile sensor pads can be easily integrated onto various robotic manipulators, including the surfaces of robot arms. In this paper, we install the tactile sensors on a soft and adaptable fin-shaped gripper, as shown in Fig. 2(b). These flexible sensor pads bend with the soft gripper and continue to provide effective signal transmission, making the system versatile across a wide range of robotic applications.

As illustrated in Fig. 2(b), each finger of the manipulator is equipped with a sensor pad containing 256 sensing units (a 16 by 16 sensor array). The size, density, and spatial sensing resolution of the sensor pads can be customized; in our current design, the resolution is set at $3mm^2$ per sensor point. Similar to [7], 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. These layers are then encapsulated between two shaped Polyimide films using a high-strength adhesive (3M 468MP), ensuring robust electrical contact between the electrodes and the piezoresistive film, which is crucial for reliable signal acquisition. The sensor characteristics are repeatable across multiple devices and reliable over multiple cycles. More detailed information on tactile sensor manufacturing is available in the Appendix.

The resistance of the piezoresistive layer changes in response to applied pressure, enabling each sensor point to convert mechanical pressure into an electrical signal. These analog signals are captured by




(a) Real World Environment (b) 3D Data Processing (c) Decision-Making Module

Figure 3: **Visuo-Tactile Policy. Part (a)** shows the real-world setup and the manipulated objects. **Part (b)** illustrates the processing of visual data (upper block) and tactile data (bottom block), followed by their integration within the same 3D coordinates. From the visualization of the tactile signals, depending on the relative movements of the two grippers, the force patterns on the two fingers of the same gripper can differ even when grasping a symmetric part of the tool. Such nuanced information is particularly important for in-hand object manipulation. **Part (c)** outlines our decision-making process, where our network takes the integrated 3D visuo-tactile representations as input and outputs the predicted action sequence.

an Arduino Nano and transmitted to a computer via serial communication. We use a customized electrical readout circuit to acquire data at a frame rate of up to approximately 32.2 FPS. The total cost of one sensor pad and the reading board (without Arduino) is about $20. **We are committed to releasing comprehensive tutorials for hardware manufacturing.**

**Integration of Flexible Tactile Sensor and Soft Gripper.** We install our tactile sensors on the surface of a fully 3D-printed soft gripper made from TPU material (Fig. 2(b)). The tactile sensor pads integrate well with the flexible soft gripper. Our new gripper design offers several advantages. First, the soft nature of the gripper significantly increases the contact area between the sensors and the target objects. This not only helps stabilize the manipulation process but also ensures consistent reflection of the contacting patterns and geometry of the objects. Second, while our visuo-tactile policy provides certain levels of action compliance, the softness of the gripper adds mechanical compliance [2], enabling us to handle fragile objects more effectively.

## 3 Learning Visuo-Tactile Dexterity

### 3.1 Problem Formulation

In our study, we address the challenge of learning contact-rich robot skill trajectories through imitation learning. Fig. 3 provides an overview of the integration of visuo-tactile data and the subsequent action generation processes. Specifically, we introduce a visuo-tactile policy, denoted as $\pi : \mathcal{O} \rightarrow \mathcal{A}$. This policy maps combined visual and tactile observations $o \in \mathcal{O}$ to actions $a \in \mathcal{A}$. Our method consists of two critical parts: **(1) Dense Visuo-Tactile Representation:** Fig. 3(b) shows the integration of visual and tactile data within a unified coordinate system, which includes: (i) 3D Visual Point Cloud (visualized by •): Captured by the camera, formatted as $P_t^{\text{visual}} \in \mathbb{R}^{N_{\text{vis}} \times 4}$, including an additional empty channel to match the shape of the tactile data. (ii) 3D Tactile Point Cloud (visualized by •): This tactile point cloud includes all the points of the tactile sensing units and uses the sensing value as a feature channel, formatted as $P_t^{\text{tactile}} \in \mathbb{R}^{N_{\text{tac}} \times 4}$. **(2) Policy Learning:** Fig. 3(c) indicates the imitation learning process. Conditioned on our 3D dense visuo-tactile representation, we leverage the diffusion policy [1] to generate actions as a sequence of robot joint states.

### 3.2 Dense Visuo-Tactile Representation

In our approach, instead of separately processing tactile and visual modalities for feature extraction [5], we integrate tactile and visual data by projecting them into the same 3D space. As illustrated in Fig. 3(b), the top row demonstrates the processing of visual observations, while the bottom row depicts the processing of dense 3D tactile points using tactile signals and robot proprioception.

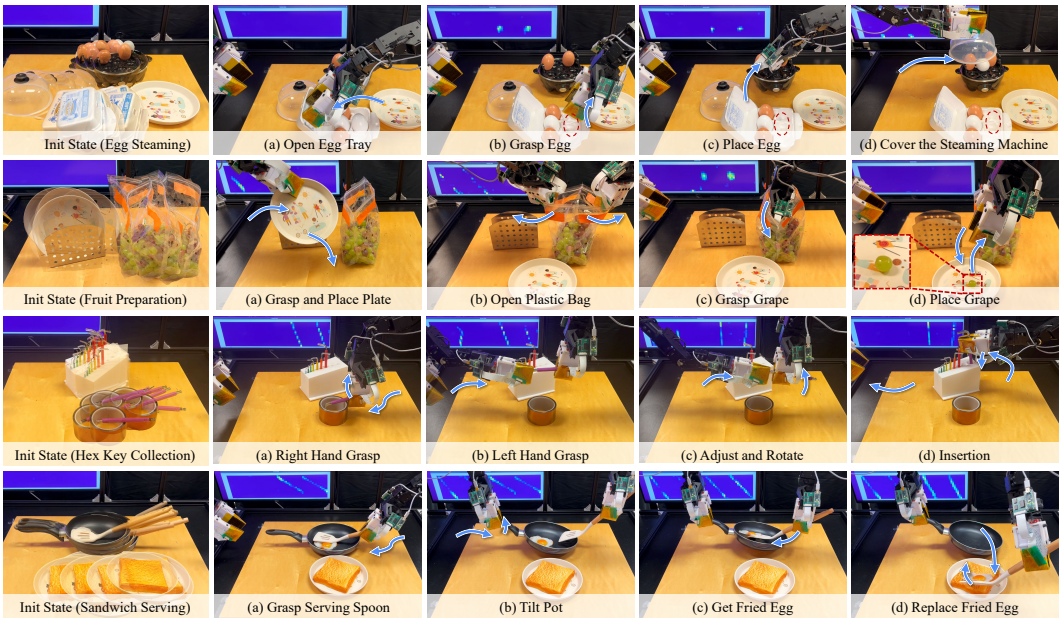



Figure 4: **Policy Rollout.** We evaluate our visuo-tactile policy across four long-horizon, precise manipulation tasks. Detailed descriptions and metrics for these tasks can be found in Sec. **??**. The first two rows emphasize tasks that require fine-grained force information, while the last two rows focus on tasks that require in-hand object state information. Please check videos on our website for more details.

Tasks Requiring Fine-Grained Force Information

| **Modalities** | **Egg Steaming (30 demos)** | | | | **Fruit Preparation (30 demos)** | | | |
|---|---|---|---|---|---|---|---|---|
| | Open Egg Tray | Grasp Egg | Place Egg | Whole Task | Grasp and Place Plate | Open Plastic Bag | Grasp Grapes | Whole Task |
| RGB Only | 0.95 | 0.75 | 0.60 | 0.50 | 0.95 | 0.75 | 0.45 | 0.45 |
| RGB w/ Tactile Image | 0.95 | **0.90** | 0.80 | 0.70 | 0.90 | 0.80 | 0.70 | 0.70 |
| PC. Only | **1.00** | 0.75 | 0.65 | 0.55 | **1.00** | 0.85 | 0.50 | 0.45 |
| PC. w/ Tactile Image | **1.00** | 0.85 | 0.70 | 0.70 | 0.95 | **0.90** | 0.75 | 0.65 |
| PC. w/ Tactile Points (Ours) | **1.00** | **0.90** | **0.90** | **0.85** | 0.95 | 0.85 | **0.80** | **0.80** |

Tasks Requiring In-Hand State Information

| **Modalities** | **Hex Key Collection (30 demos)** | | | | **Sandwich Serving (50 demos)** | | | |
|---|---|---|---|---|---|---|---|---|
| | Right Hand Grasp | Left Hand Grasp | In-Hand Adjustment | Whole Task | Grasp Serving Spoon | Tilt Pot | Get Fried Egg | Whole Task |
| RGB Only | 0.90 | 0.85 | 0.55 | 0.45 | **1.00** | 0.95 | 0.70 | 0.60 |
| RGB w/ Tactile Image | 0.95 | 0.90 | 0.70 | 0.60 | **1.00** | **1.00** | 0.85 | 0.75 |
| PC. Only | **1.00** | 0.90 | 0.65 | 0.65 | **1.00** | **1.00** | 0.75 | 0.65 |
| PC. w/ Tactile Image | 0.95 | 0.85 | 0.60 | 0.50 | **1.00** | 0.90 | 0.75 | 0.65 |
| PC. w/ Tactile Points (Ours) | **1.00** | **0.95** | **0.95** | **0.90** | **1.00** | **1.00** | **0.90** | **0.85** |

Table 1: **Comparison with Baselines.** We evaluate our policy over 20 episodes and the best performance for each task is bolded. Please check our website for more comparison videos.

### 3.3 Training Procedure

As shown in Fig. 3(c), the decision module in our method is formulated as a conditional denoising diffusion model [4]. It uses the PointNet++ [6] architecture as the backbone and is conditioned on the 3D visuo-tactile representation $o$ to denoise random Gaussian noise into the actions $a$.

## 4 Experiments

In the experiments, we primarily compare our methods with the following baselines. We trained all policies for 2,000 epochs. All methods, including ours and the baselines, use three camera views.
*(1) RGB Only*. This method uses multi-view RGB from cameras as input for the image-based diffusion policy. We use the same implementation as [1].
*(2) RGB w/ Tactile Image*. This method processes multi-view RGB images and tactile images through different branches for the diffusion policy. We use a CNN as the feature extractor for tactile images.
*(3) PC. Only*. This method uses only multi-view visual point clouds as the sensing modality for the diffusion policy. We use PointNet++ as the feature extractor.
*(4) PC. w/ Tactile Image*. This method fuses multi-view visual point clouds and tactile images through different branches for the diffusion policy. We use a CNN as the feature extractor for tactile images and PointNet++ as the feature extractor for point clouds.

## References


[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[2] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[3] 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.

[4] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[5] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[7] S. M. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, and W. Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569:698 – 702, 2019.

## Contents


**1 Introduction** **1**

**2 Visuo-Tactile Manipulation System** **2**

2.1 Sensor and Gripper Design . . . . . 2

**3 Learning Visuo-Tactile Dexterity** **3**

3.1 Problem Formulation . . . . . 3

3.2 Dense Visuo-Tactile Representation . . . . . 3

3.3 Training Procedure . . . . . 4

**4 Experiments** **4**

**A Discussion and Comparison** **6**

**B Single Arm Experiments** **7**

**C Tactile Sensor Hardware** **7**

C.1 Tactile Sensor Manufactory . . . . . 7

C.1.1 Tactile Sensor Pad Design . . . . . 7

C.1.2 Reading Board Design . . . . . 7

C.2 Tactile Hardware Evaluation Experiment . . . . . 9

C.2.1 Physical Characteristics . . . . . 9

C.2.2 6-DoF Object Pose Estimation . . . . . 9

**D Experiment details for Imitation Learning** **10**

D.1 System Overview . . . . . 10

D.2 Experiment Setup Details . . . . . 10

D.2.1 Details for the Egg Steaming task . . . . . 11

D.2.2 Details for the Fruit Preparation Task . . . . . 12

D.2.3 Details for the Hex Key Collection Task . . . . . 12

D.2.4 Details for the Sandwich Serving Task . . . . . 13

D.3 Learning Details . . . . . 13


# A Discussion and Comparison

While our community has highlighted the advantages of tactile sensors, there are still gaps when it comes to scalable, low-cost tactile sensors. Compared to GelSight, our sensors obtain advantages in compactness and flexibility, offering the balance between form factor constraints and sensing resolution:

**(a) Thinness:** Our sensor is no more than 1 mm as thin as paper. This feature enables robots to perform precise manipulations in narrow spaces. For instance, one of our tasks involves grasping grapes inside a bag, where the robot needs to insert its finger between the gaps of grapes to grasp them. Achieving this is nearly impossible with the thickness of GelSight sensors. **(b) Flexibility:** Our sensor is soft and flexible, unlike GelSight. This flexibility allows it to be installed on soft grippers,

enhancing the contact area through mechanical compliance. This not only makes the contact pattern more informative but also stabilizes the grasp, as proven in the UMI study [2]. **(c) Extremely high Resolution may not be needed:** While optical-based tactile sensors have extremely high resolution, often surpassing human capabilities, such high resolution may not be necessary for everyday tasks. According to the Two-Point Discrimination Test, the human fingertip's resolution is about 2.5 mm for point localization and 5 mm for the two-point threshold (the minimal distance to differentiate two points on the skin). The resolution of our sensor design is partly inspired by human tactile sensing capabilities, which can be sufficient for many practical applications while keeping the sensor thin and flexible enough.

# B Single Arm Experiments

**Light Bulb Retrieval:** (i) Rotate. The robot locates the light bulb, rotates it, and retrieves it from the base. (ii) Relocation. The robot places the retrieved light bulb carefully on the table.

**Peg Insertion:** (i) Grasp. The robot grasps a peg from the table. (ii) Insertion. The robot inserts the peg into a hole.

| | Single-Arm Tasks | | | |
|---|---|---|---|---|
| **Modalities** | **Light Bulb Retrieval (30 demos)** | | **Peg Insertion (30 demos)** | |
| | Rotation | Relocation | Grasp | Insertion |
| RGB Only | 0.85 | 0.85 | 1.00 | 0.30 |
| RGB w/ Tactile Image | 0.95 | 0.95 | 1.00 | 0.65 |
| PC. Only | 0.85 | 0.85 | 1.00 | 0.65 |
| PC. w/ Tactile Points (Ours) | **1.00** | **1.00** | **1.00** | **0.85** |
| Ours (Using a New Set of Sensors) | **1.00** | **1.00** | / | / |
| Ours (Changing Sensor Location Slightly) | 0.95 | 0.90 | / | / |

Table 2: **Comparison with Baselines.** We evaluate our single-arm policy over 20 episodes and the best performance for each task is bolded. Our sensors are repeatable across devices after normalization. Even though the reading may have slight variances among different sensors, the force pattern and distribution are almost identical after normalization.

# C Tactile Sensor Hardware

## C.1 Tactile Sensor Manufactory

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

After obtaining the tactile sensor pad, we attach the sensors to the robot fingers. The order of each tactile sensor unit is visualized in Fig. 7. We clearly define the tactile order to ensure that each sensor's position can be accurately calculated, and the tactile signals can correctly correspond to our real setting and dataset.

### C.1.2 Reading Board Design

To ensure easy installation of the tactile reading board in the robot, we have designed it to be as compact as possible, as shown in Fig. 6. The tactile reading board measures 45.5 mm $\times$ 48.4 mm and includes an Arduino. The small size further enhances the scalability of our tactile sensors. We

Figure 5: **Tactile Physical Characteristics Evaluation Experiment**

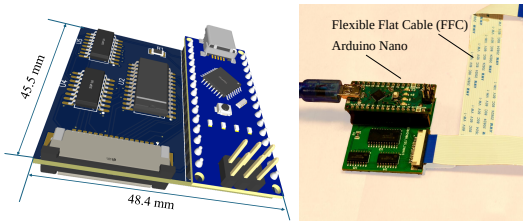



Figure 6: **Tactile Reading Board Design.**

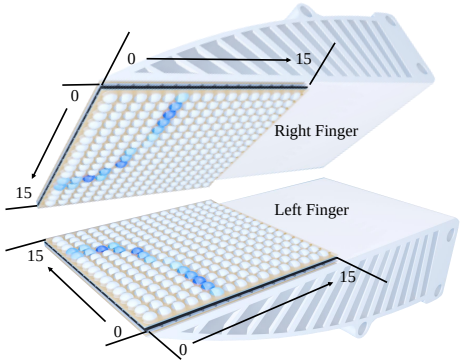
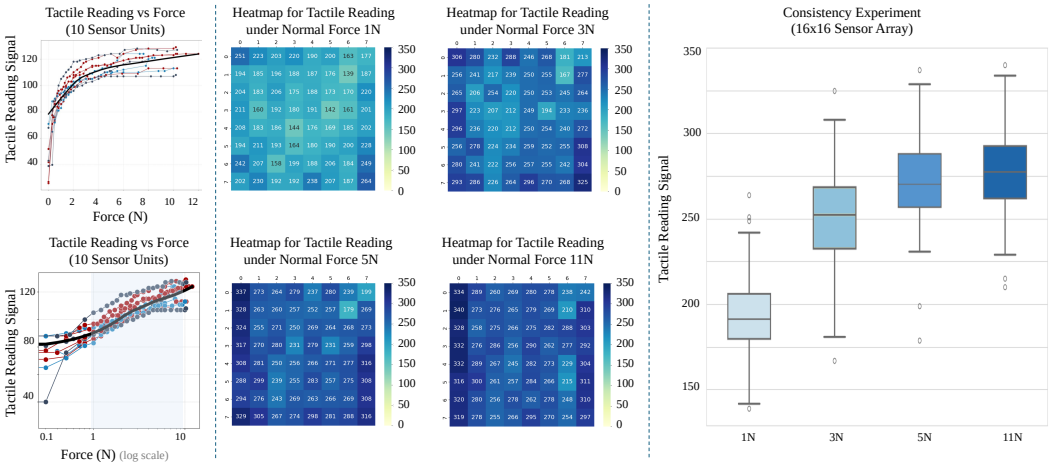



Figure 7: **Tactile Sensor Order Visualization.**




(a) Individual Sensors Test (b) Heatmap for Tactile Pad Reading under Different Loads (c) Box Plot for Tactile Pad under Different Loads

Figure 8: **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 $8 \times 8$ 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.

use two 8-bit shift registers and one 16-channel analog switch to process the tactile signals, which are then input to the Arduino. The ADC in the Arduino converts the analog signals from the tactile sensor into digital signals and forwards them to the host via serial communication. We will release a comprehensive reading board scheme so that the community can directly order from a PCB supplier to easily replicate our tactile sensor.

### C.2 Tactile Hardware Evaluation Experiment

#### C.2.1 Physical Characteristics

To investigate the physical characteristics of our tactile sensors, we designed two experiments. As illustrated in Fig. 5, we use a force gauge to apply specific force on the tactile sensor surface. The first experiment tests how individual tactile sensor units react to applied force. The second experiment aims to test the consistency of the entire sensor pad, showcasing the variance between different regions on the tactile sensor pad.

**Individual Sensor Performance.** We began by randomly selecting 10 sensors from a total of 256 sensors in one sensor pad. For each selected sensor, we applied a normal force incrementally, ranging 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. 8 (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. 8 (a).

**Tactile Sensor Pad Consistency.** In the second part of the experiment, we used a $16 \times 16$ tactile sensor pad and divided it into $8 \times 8$ blocks, with each block comprising 4 sensors ($2 \times 2$ matrix area). Uniform loading was applied across each block using the force gauge with a circular contacting area of $176.7\ \text{mm}^2$. For each $2 \times 2$ 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. 8 (b). We calculated the mean and standard deviation for these data points and removed outliers. Finally, as illustrated in Fig. 8 (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.

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

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

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

We solve the pose-tracking problem using particle filtering [3]. We first define our observation function $P^{\text{obs}} = f(\mathbf{T})$ and then the weighting functions $w = g(P^{\text{obs}}, P^{\text{tactile}})$ as follows:

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

The observation function transforms the object model point cloud using $\mathbf{T}$, while the weighting function calculates the distance from the contact points to the observation points. In practice, we

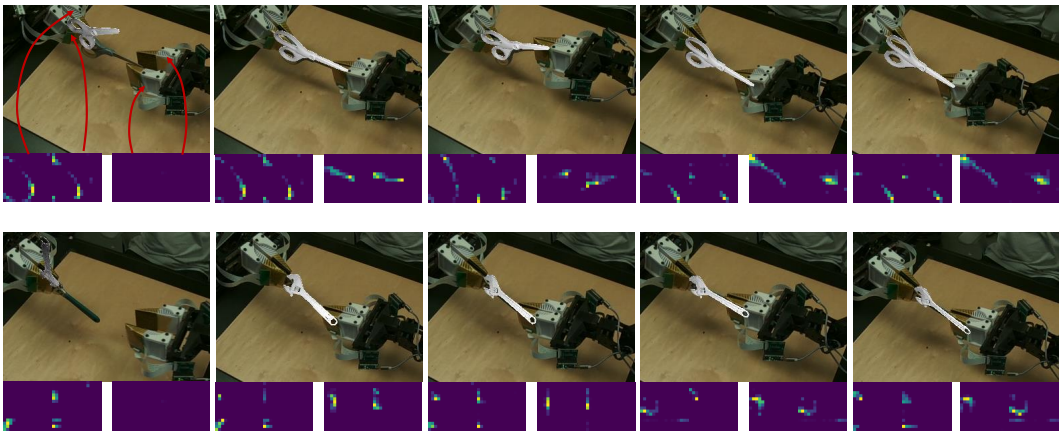


Figure 9: **Pose Estimation.** In this experiment, we estimate the object pose **without** vision information. We can see that our pose estimation becomes more accurate as we have more complete tactile signals. We can also track the object's pose as it rotates. Through this estimation, we demonstrate that our hardware can be potentially used for in-hand pose estimation and other visuotactile tasks.

scale the weights using an exponential function to facilitate convergence. Given the observation and weighting functions, we employ a standard particle filter to determine the object's pose.

Some example results are shown in Figure 9. Before the right-side robot makes contact with the object, we can only rely on the tactile signals from the left-side robot. Therefore, there are a lot of plausible solutions. Although our estimated pose is one of the plausible solutions given the one-side tactile signal, the estimation is still inaccurate. When the right-side robot contacts the object, our estimated pose aligns well with visual observation. Also, when the object is rotating in the hand, the object pose is tracked accurately.

# D Experiment details for Imitation Learning

## D.1 System Overview

As shown in Fig. 10, We employ a bimanual teleoperation system with three Realsense cameras and four tactile sensor pads on four robot fingers. Our tactile signal communication is facilitated by a multi-threaded ROS (Robot Operating System) node. This node captures tactile signals and publishes them at a frequency of 30 Hz. All data, including that from cameras and tactile sensors, is collected through multi-threading. Each data frame received is timestamped, and after an episode is completed, we align all data with these timestamps. This synchronization is crucial for maintaining the consistency of the multimodal dataset, enabling accurate temporal alignment between tactile feedback and visual data. To manage the heavy load of processing frames from three cameras, we collect data at 10 Hz to ensure consistency. We set a top camera (Realsense 455) to cover the entire workspace and positioned two other cameras (Realsense 435) close to the workspace to capture more detailed information. When using point cloud data from multiple cameras, we incorporate data from all cameras. For the baseline method using a single camera, we use only the top camera.

We also implement real-time tactile information feedback, as shown in Fig. 10 (a). During data collection, tactile signals are visually displayed on the operator's screen, enabling them to assess the adequacy of contact for secure grasping. Additionally, during the policy rollout, this visualization helps us see in real-time how tactile information relates to robot motion.

## D.2 Experiment Setup Details

In this section, we discuss the detailed information of the four tasks described in the main paper. Each task consists of four steps, as illustrated in Fig 12. We will discuss the motions and evaluation metrics for each step, and highlight how these steps demonstrate the capabilities of our tactile sensors. The typical failure cases are shown in Fig. 11 and will be discussed in the following sections.

Figure 10: **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.

#### D.2.1 Details for the Egg Steaming task

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

*Step 2: Grasp Egg.* The robot uses its right hand to grasp the egg in the tray. This motion is complex, requiring the robot to slowly increase the force and carefully grasp the egg despite heavy occlusion. The robot with a visuo-tactile policy will retry if there is no stable tactile signal in hand, while a vision-only policy may proceed to the next goal due to heavy occlusion as shown in Fig. 11 (a). *Evaluation Metrics:* The robot can reattempt to grasp the egg, but the step fails if it moves to the next stage without the egg or if the egg falls during the transition from the tray to the steaming machine. Additionally, prolonged time spent in the egg tray will also be considered a failure.

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

*Step 4: Cover the Steaming Machine.* The robot needs to use its left hand to grasp the cover of the steaming machine and place it safely inside. This task is challenging due to the unique shape of the

steaming machine's handle, as shown in Fig. 11 (a). The robot must apply a precise amount of force to the handle: sufficient to lift it but not so much that the cover flips and falls to the ground. The robot must apply a precise amount of force to the handle: sufficient to lift it but not so much that the cover flips and falls to the ground. This step showcases how our tactile sensor enables the robot to perform fine-grained grasping manipulations, similar to a human's ability to apply suitable and stable force to grasp objects. *Evaluation Metrics:* The robot is allowed multiple attempts to grasp the cover. The task is considered successful if the cover is securely placed on the steaming machine. It is considered a failure if the cover flips or falls during the process.

#### D.2.2 Details for the Fruit Preparation Task

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

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

*Step 3: Grasp the fruit.* The robot needs to use its right hand to get inside the plastic bag and grasp the fruit. This step is the most important and difficult in this task. First, as shown in Fig. 12(Task 2: Fruit Preparation), the robot and manipulated objects are highly occluded in the bag, making it impossible for visual information to observe critical details. Our visuo-tactile policy will grasp multiple times until there is stable tactile information to secure the grapes, while a vision-only policy typically attempts the motion once regardless of the presence of grapes, making the grasping success a random event. Second, the grapes are usually clustered together, requiring the robot to apply suitable force to avoid damaging the fruit. Our visuo-tactile policy can successfully grasp single or multiple grapes from the bag, while a vision-only policy may break the grapes when the robot grasps multiple grapes (as shown in Fig. 11 (b)). since it aligns the gripper joint states instead of using force-related information. Third, this task also showcases our sensors' human-like dexterous manipulation; our tactile-integrated gripper is thin enough to get into the gaps between grapes, making it easier to grasp the grapes in a cluster. *Evaluation Metrics:* The task is considered successful when the robot successfully grasps the fruit out of the plastic bag. The task is considered a failure if the robot breaks the grapes or moves to the next stage without the grapes. The policy also fails if the robot stays in the bag for a long time without moving, which usually happens with the vision-only policy that is confused about the states of the objects and the robot end-effector under high occlusion.

*Step 4: Place grape.* The robot needs to place the grapes on the plate. The task may fail if the robot uses too much force to grasp the grapes, causing them to stick in the gripper and resulting in failure. *Evaluation Metrics:* The task is considered successful if the robot successfully places the grapes on the plate and returns to the initial position.

#### D.2.3 Details for the Hex Key Collection Task

*Step 1: Right Hand Grasp.* The robot needs to use its right hand to grasp the tail of the hex key and lift it stably to the middle of the air. The initial position of the hex key is tricky, but it reflects a common daily life scenario where only the tail of the hex key is accessible, requiring additional adjustments to insert the hex key properly. A typical failure case of baselines, shown in Fig. 11 (c), occurs when the robot does not secure a stable grasp, resulting in significant slippage during the lifting process. Even if the hex key remains in-hand, this slippage can cause subsequent task failures. One observation during the experiment is the consistent small slippage during the first grasp, leading to variations in the hex key's in-hand pose, which adds complexity to the following steps. *Evaluation Metrics:* The robot successfully grasps the hex key without significant slippage.

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

*Step 3: In-hand Adjustment.* The robot's left and right hands need to cooperate to adjust the hex key's position so that it is in a ready pose for the following insertion. Our goal is to adjust the hex key to be

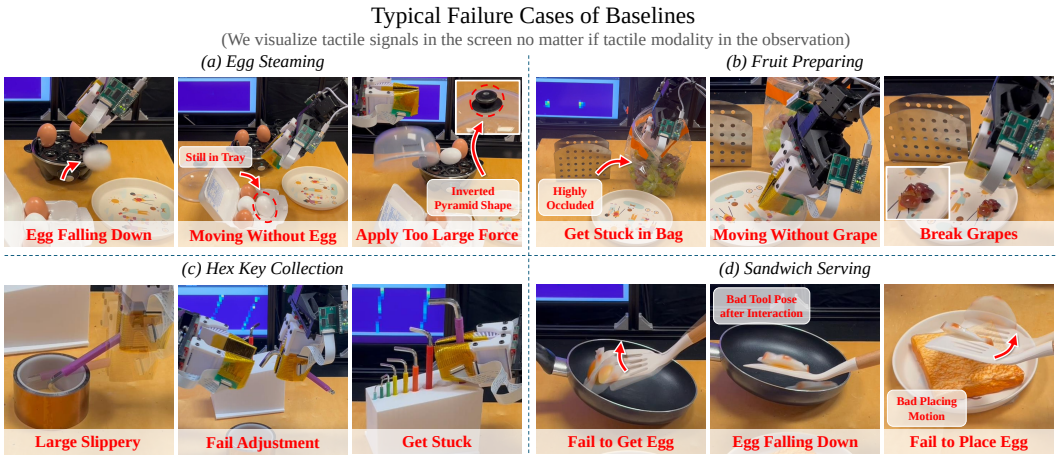



Figure 11: **Failure Cases.** We present typical failure cases of the baseline method for all four tasks and analyze the reasons for these failures to highlight the complexity of the tasks and the importance of tactile feedback during these steps.

perpendicular to the robot's fingers, making the subsequent insertion task easier. The vision-only policy usually fails to adjust the position correctly, shown in Fig 11(c), making the following insertion impossible. *Evaluation Metrics:* The robot's two arms must cooperate to adjust the hex key's pose. The final pose should have a sufficiently long tail, and the hex key should be almost perpendicular to the robot's fingers.

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

#### D.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.

### D.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:

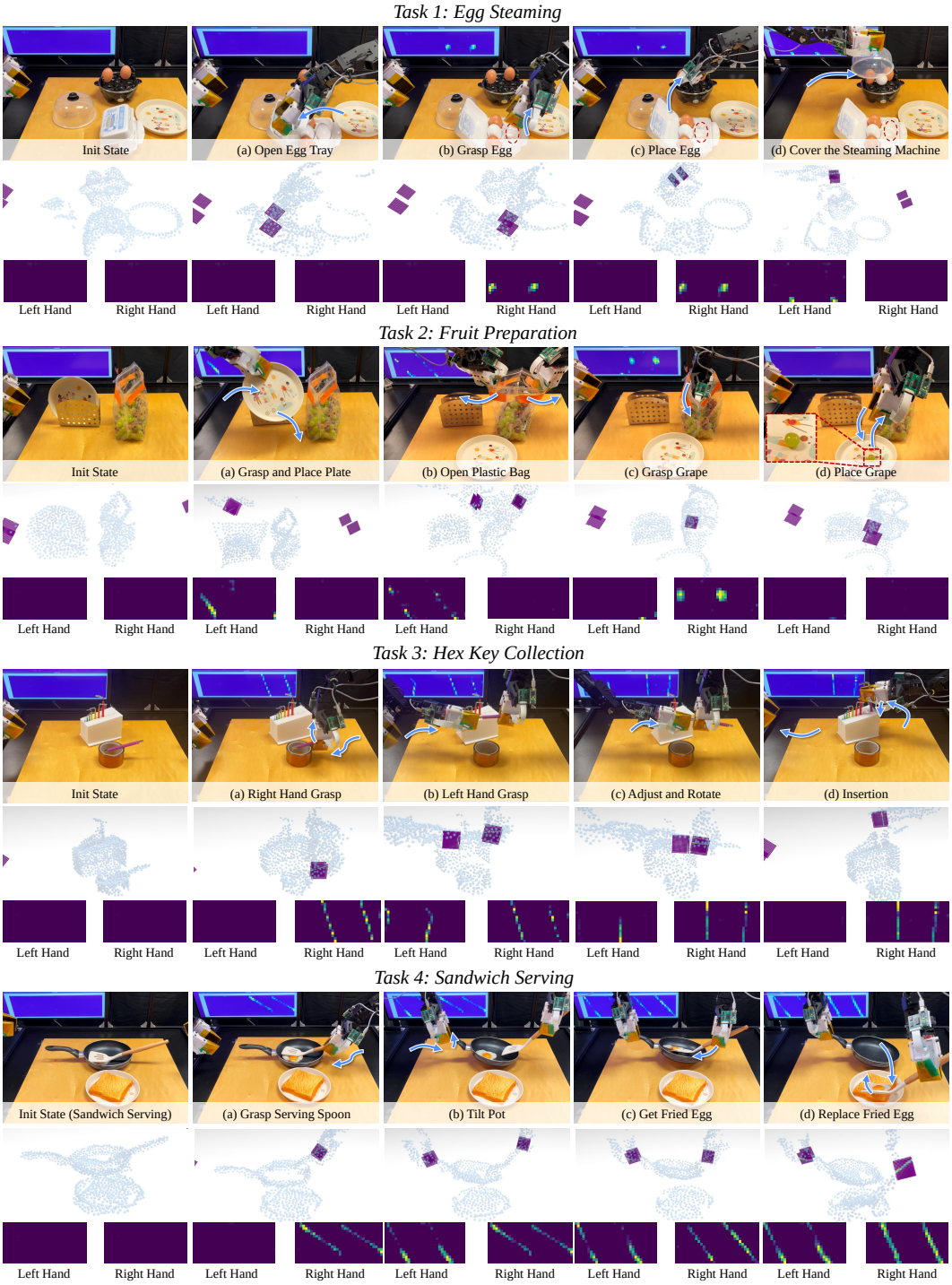



Figure 12: **Quantative Result of Tactile Representation.** Here we showcase a total of four tasks. For each task, the first row presents the real image. In the second row, we visualize our visuo-tactile points in a unified 3D space to demonstrate how tactile points can infer spatial relationships between objects and contact areas. The third row provides a 2D image to clearly visualize the tactile signals.

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.