# FingerEye: Continuous and Unified Vision-Tactile Sensing for Dexterous Manipulation

Zhixuan Xu[1,2], Yichen Li[1,2], Xuanye Wu[2,3], Tianyu Qiu[2,4], Lin Shao[1,2†]
[1] National University of Singapore [2] RoboScience
[3] Huazhong University of Science and Technology [4] South China University of Technology
[†] Corresponding author
https://nus-lins-lab.github.io/FingerEyeWeb/

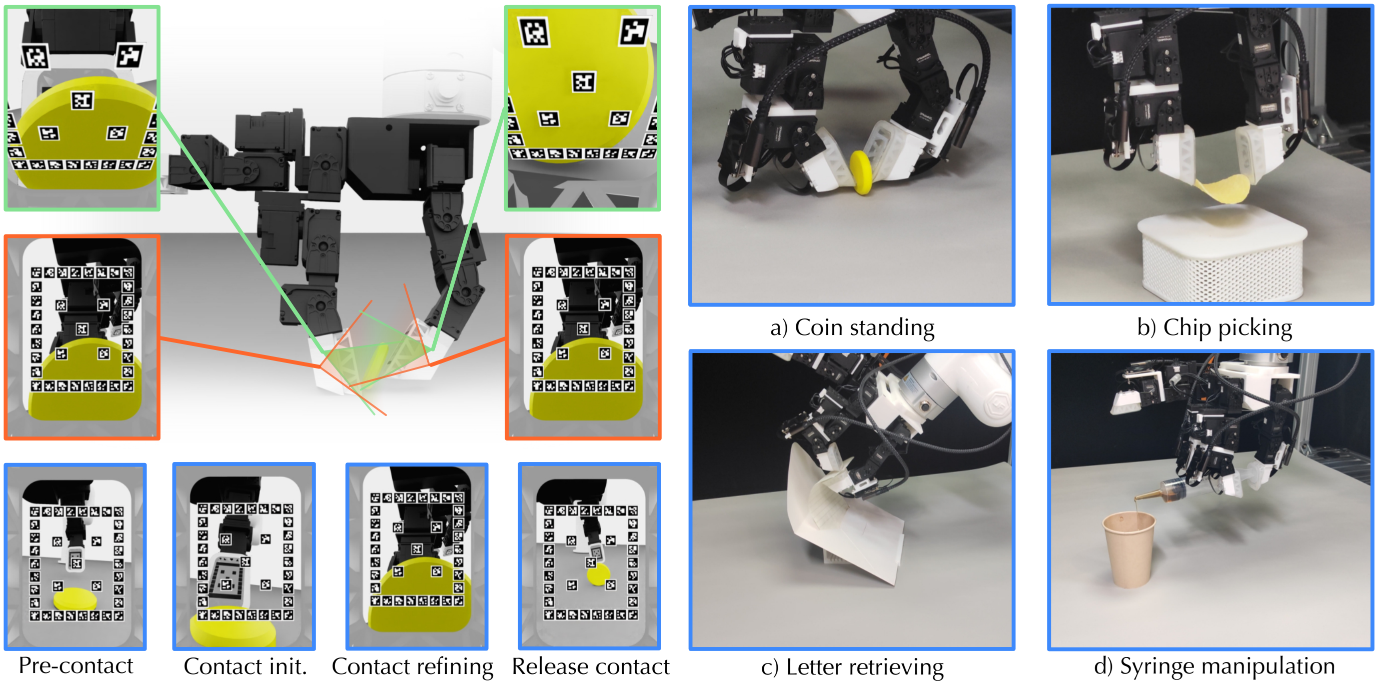


Fig. 1: **FingerEye overview and capabilities.** FingerEye provides continuous vision-tactile perception across all phases of interaction. Before contact, binocular RGB cameras provide close-range visual cues and implicit stereo depth to guide fingertip positioning. Upon contact, external forces and torques deform a compliant ring structure; marker-based pose estimation converts these deformations into contact wrench signals for contact monitoring and control. *Left*: the simulation digital twin of the FingerEye platform. *Right*: real-world evaluation tasks spanning diverse object properties and interaction regimes, including *Chip Picking* (rigid, delicate), *Coin Standing* (rigid, contact-sensitive), *Letter Retrieving* (deformable, thin-shell), and *Syringe Manipulation* (articulated, functional).

***Abstract*—Dexterous robotic manipulation requires comprehensive perception across all phases of interaction: pre-contact, contact initiation, and post-contact. Such continuous feedback allows a robot to adapt its actions throughout interaction. However, many existing tactile sensors, such as GelSight and its variants, only provide feedback after contact is established, limiting a robot's ability to precisely initiate contact. We introduce FingerEye, a compact and cost-effective sensor that provides continuous vision-tactile feedback throughout the interaction process. FingerEye integrates binocular RGB cameras to provide close-range visual perception with implicit stereo depth. Upon contact, external forces and torques deform a compliant ring structure; these deformations are captured via marker-based pose estimation and serve as a proxy for contact wrench sensing. This design enables a perception stream that smoothly transitions from pre-contact visual cues to post-contact tactile feedback. Building on this sensing capability, we develop a vision-tactile imitation learning policy that fuses signals from multiple FingerEye sensors to learn dexterous manipulation behaviors from limited real-world data. We further develop a digital twin of our sensor and robot platform to improve policy generalization. By combining real demonstrations with visually augmented simulated observations for representation learning, the learned policies become more robust to object appearance variations. Together, these design aspects enable dexterous manipulation across diverse object properties and interaction regimes, including coin standing, chip picking, letter retrieving, and syringe manipulation. The hardware design, code, appendix, and videos are available on our project website at https://nus-lins-lab.github.io/FingerEyeWeb/.**

## I. Introduction

"Manipulation refers to an agent's control of its environment through selective contact." [1] Achieving such selective contact in dexterous robotic manipulation requires sensing that remains informative throughout the entire interaction process: before contact, during contact initiation, and after contact. Consider the task of standing a coin upright on a table, as shown in Fig. 1. Before contact, the robot must visually localize the thin coin edge and refine its approach in a closed-loop manner with millimeter-level precision, as even slight unintended contact can push the coin away. During contact initiation, the robot must detect the exact instant at which each fingertip first touches the coin, since premature or asymmetric contact can easily destabilize it. After contact is established, the robot must continuously monitor the contact state and regulate applied forces to keep the coin balanced without slipping or toppling. This example highlights the need for continuous sensing across interaction phases, enabling precise contact point and force regulation in dexterous manipulation.

Despite its importance, achieving such comprehensive sensing remains challenging. Most robotic manipulation systems rely on external vision from environment-mounted or wrist-mounted cameras together with fingertip tactile sensing [2, 3]. External vision provides global scene context but offers only weak and indirect cues about contact, which are often subtle and easily occluded near the interaction regions. Tactile sensors, including vision-based tactile sensors such as GelSight [4], 9DTact [5], provide rich local contact information but only after physical contact has been established. As a result, vision and touch are often treated as phase-specific signals, with vision used before contact and tactile sensing afterward. However, contact initiation is a continuous process that requires anticipating contact, maintaining visual alignment, and regulating control as forces begin to emerge. When vision and tactile sensing are decoupled, the system lacks a continuous signal to indicate and guide this transition, leading to unstable contact initiation and brittle control in dexterous manipulation.

Works [6, 7] on See-Through-Skin sensors aim to bridge vision and touch by integrating visual and tactile perception within a single sensing surface. While these approaches reduce the gap between pre-contact and post-contact sensing, they typically rely on semi-transparent materials [8, 9, 10, 11], illumination switching [12, 13], or mechanically reconfigurable components [14, 15], and are often constrained to a central frontal contact region. Moreover, inherent trade-offs between depth perception, sensor form factor, and sensing fidelity limit their effectiveness in confined spaces and in scenarios involving peripheral or non-frontal contact.

To address these limitations, we present FingerEye, a sensing system that integrates binocular RGB cameras with a compliant soft ring surrounding a transparent acrylic cover carrying a custom AprilTag layout. The cameras provide close-range fingertip-centric visual observations for contact localization, while the compliant ring deforms under external forces and torques. These deformations are captured via marker-based pose estimation using the same visual sensing pipeline.

Compared to vision-based tactile sensors such as GelSight [4] and TacTip [16], which primarily sense deformation after contact FingerEye provides a *single, continuous sensing stream* that remains valid across all interaction phases. The same visual observations encode object geometry before contact and contact-induced deformation after contact, without modality switching or separate sensing pipelines. In contrast to recent See-Through-Skin (STS) sensors, FingerEye adopts a compact wedge-shaped form factor tailored for dexterous fingertips. A binocular camera system with complementary placements and focal configurations provides implicit depth perception across interaction phases. A compliant soft ring surrounding a transparent acrylic cover supports both frontal and peripheral contact, expanding the effective sensing region in confined or cluttered spaces. A custom AprilTag layout enables reliable marker-based pose estimation and sensitivity to subtle 6D contact wrenches. The design relies only on low-cost, off-the-shelf components and simple 3D-printed parts, making it substantially easier to reproduce than multi-layer or illumination-heavy STS sensors. By avoiding colored or active illumination and heuristic sensing-mode switching, FingerEye produces natural RGB observations throughout interaction, reducing visual domain gaps and enabling stable perception and control in contact-rich dexterous manipulation tasks.

Building on this hardware foundation, we further develop a robot and data collection platform, digital twin, and learning framework to support efficient learning of dexterous manipulation skills. In summary, our primary contributions are:

- **A continuous and unified vision-tactile sensing paradigm.** We introduce **FingerEye** which provides continuous and unified vision-tactile sensing spanning pre-contact, contact initiation, and post-contact interaction.
- **A mechanical sensor design tailored for dexterous manipulation.** FingerEye adopts a compact wedge-shaped form factor and delivers low-domain-gap RGB observations with implicit depth perception, enables detection of subtle contact wrenches from both frontal and peripheral regions. The sensor is fully reproducible using low-cost, off-the-shelf, and 3D-printed components.
- **An integrated platform for data collection and scaling.** We develop a data collection system that supports intuitive real-and-sim demonstration collection with multiple **FingerEye** sensors. We develop a digital twin that mirrors the sensor's geometry and sensing characteristics, enabling scalable data generation and sim-to-real learning.
- **A learning framework for dexterous manipulation.** We propose **FingerEye Policy**, which fuses multiple **FingerEye** sensors and leverages simulation-augmented representation learning to efficiently learn robust behaviors from limited real-world data.
- **An extensive experimental evaluation.** We conduct comprehensive experiments across diverse contact-rich manipulation tasks to validate the proposed sensing system and learning framework.

Our hardware design, code, appendix, and videos are available on our project website at https://nus-lins-lab.github.io/FingerEyeWeb/.

## II. Related work

We review prior work on vision-based tactile sensing and See-Through-Skin (STS) sensors. For a broader review of the field, we refer to [17, 18, 19]. Robotic manipulation has motivated a wide range of tactile sensor designs. Vision-based tactile sensors, such as GelSight and its variants [4, 20, 21, 22, 23, 24], capture high-resolution images of soft material deformation and provide detailed information about contact geometry, pressure distribution, and surface texture. While highly informative after contact is established, these sensors provide no feedback during the pre-contact phase, leaving robots without sensing cues prior to physical interaction.

See-Through-Skin sensors address this limitation by combining visual and tactile perception within a single sensing surface. These designs typically replace opaque membranes with semi-transparent gels or elastomers [25] and recover different sensing modalities through alternating internal illumination [7, 26, 12] or mechanically reconfigurable components [27]. Our concurrent work, TacThru [8], follows this direction by using a transparent elastomer and persistent illumination to eliminate explicit mode switching, at the cost of explicit depth perception, and employs keyline markers to improve visibility in complex visual backgrounds.

In contrast, FingerEye adopts a compact, fingertip-scale, peripheral-contact-aware sensing architecture. A wedge-shaped form factor with a compliant silicone ring surrounding a transparent acrylic cover allows contact and deformation to arise from both frontal and peripheral regions, expanding the effective sensing area and enabling interaction in narrow or confined spaces. A binocular camera system with complementary camera placements provides fingertip-centric depth perception and reliable close-range geometry cues, improving contact localization and stability in contact-rich manipulation tasks. Empirically, the binocular configuration yields a measurable improvement in manipulation success rates.

FingerEye further differs in its marker and fabrication design. An AprilTag-based [28] layout enables robust marker-based pose estimation and sensitivity to subtle 6D contact wrenches under diverse visual conditions, without relying on colored illumination or dense marker patterns. This results in clean, consistent RGB observations and avoids heuristic sensing-mode switching. The sensor is constructed entirely from low-cost, off-the-shelf, and 3D-printed components using a simple silicone molding process, making it substantially easier to reproduce than multi-layered visuo-tactile and STS sensors.

## III. FingerEye Sensor Design

### A. Mechanical Highlights

FingerEye is designed for dexterous robotic fingertips. The key mechanical design highlights are summarized as follows:

- **Wedge-shaped, compact form factor** that fits naturally at the fingertip and enables operation in confined or cluttered environments.
- **Binocular camera system** with complementary placements and focal configurations, providing complementary visual information across interaction phases and enabling fingertip-centric depth perception, which empirically improves manipulation success rates significantly.
- **Compliant soft ring surrounding the transparent acrylic cover**, which allows contact and deformation to occur at the fingertip periphery, expanding the effective sensing region.
- **Custom AprilTag layout** that provides robust and reliable detection under diverse visual conditions, enabling accurate pose estimation and sensitivity to subtle 6D contact wrench cues.
- **Reproducibility**, relying only on low-cost, off-the-shelf, and 3D-printed parts, with a total material cost of approximately $60.

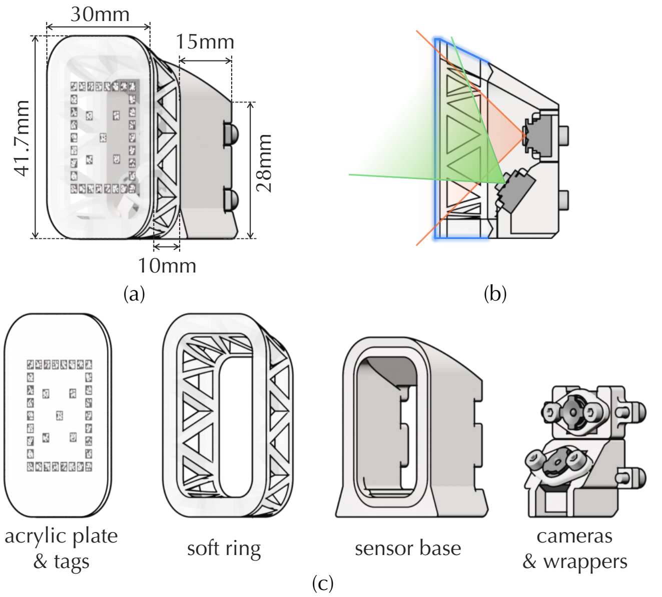


Fig. 2: **Hardware Design.** (a) Overall dimensions of the proposed vision-based tactile sensor. (b) Cross-sectional view showing the two cameras and their fields of view: the tip camera field of view (orange), the root camera field of view (green), and the frontal and peripheral contact sensing regions (blue). (c) Exploded view of the main components.

### B. Mechanical Implementation

Each FingerEye module (Fig. 2) integrates a binocular RGB camera system with a compliant soft ring, covered by a transparent acrylic plate that carries a custom AprilTag layout.

**Wedge Shape and Compact Dimensions.** The assembled FingerEye module adopts a compact wedge-shaped form factor that fits naturally at the fingertip and enables operation in narrow or cluttered environments, such as opening an envelope through a narrow mail slot. The module measures approximately 28.0 × 25.4 × 26.0 mm in overall dimensions, and detailed size is presented in Fig.2.

**Binocular Cameras.** Each FingerEye module uses two HBVCAM-2307-FPC82 V11 USB camera modules, each measuring $8.6 \times 8.6 \times 5.4$ mm with a 120° field of view. The two cameras are intentionally different in both placement and focal configuration. The *tip camera* is mounted perpendicular to the acrylic cover at a short working distance of approximately 10 mm. This configuration provides a sharp, close-up view of the AprilTag layout, enabling precise pose estimation when the soft ring deforms during contact. The *root camera* is mounted closer to the finger root and tilted by 21.48° relative to the fingertip's horizontal direction. It operates at a longer working distance of roughly 80 mm, which extends the field of view and preserves scene visibility as the hand approaches and fingertips interact with objects. Together, this binocular camera system ensures robust perception across interaction phases: the tip camera captures fine-grained local deformation during contact, while the root camera maintains broader scene awareness during approach and manipulation.

**Compliant Soft Ring.** The compliant soft ring is fabricated via silicone molding and serves both as a mechanical buffer and as a deformation-based sensing medium. Its internal structure consists of triangular elements, which introduce directional compliance: it allows controlled deformation under external forces and torques while preserving overall shape stability. Importantly, its deformation creates a consistent relationship between the applied wrench (forces and torques) and the changes in the estimated tag pose, as shown in Sec. IV-C. This validates the use of the tag pose as a reliable proxy for force and torque sensing. The soft ring surrounds and is attached to the acrylic cover. In contrast to a rigid structural ring enclosing a deformable medium such as gel, the compliant design allows contact and deformation to occur at the fingertip periphery, thereby expanding the effective sensing region beyond the central contact area.

**Acrylic Cover with a Custom AprilTag Layout.** A 1 mm-thick transparent acrylic cover is mounted above the soft ring to protect the sensing elements while preserving clear visual access to ring deformation. A custom AprilTag layout is applied to the inner surface of the cover, consisting of 35 tags from the `tag25h9` family arranged as a compact central cluster surrounded by a ring of tags to provide broad spatial coverage. Each tag measures 2 mm and includes a 0.2 mm white border to ensure reliable detection. Using many spatially distributed tags improves robustness compared with sparse layouts (e.g., a few corner markers), because all currently visible corners can be aggregated into one multi-tag PnP estimate under partial occlusion and non-uniform deformation. We analyze and evaluate pose-estimation robustness, sensitivity, and downstream manipulation utility in Sec. IV-A, Sec. IV-B, and Sec. IV-D, respectively.

**Assembly and Supporting Components.** The FingerEye module includes 3D-printed camera wrappers and M2 screws to ensure mechanical stability and optical alignment. Fabrication details and bill of materials are provided in Appendix Sec. A.4.

### C. Design Trade-offs

**Rigid transparent interface vs. contact friction.** Using a rigid acrylic cover improves optical clarity and long-term imaging stability for unified vision-tactile sensing, but it can reduce contact conformity and friction compared with softer gel interfaces. In our setup, this trade-off is mitigated by adding high-friction gripping material on camera-unobservable regions (e.g., perimeter areas and regions behind tags), which preserves visual tracking quality while improving grasp reliability.

**Passive lighting vs. active self-illumination.** We intentionally use passive lighting rather than active or colored self-illumination to keep the RGB stream consistent for both sensing and policy learning. Together with contrast enhancement (CLAHE), this design maintains reliable tag detection across low-light conditions without introducing illumination-specific color artifacts.

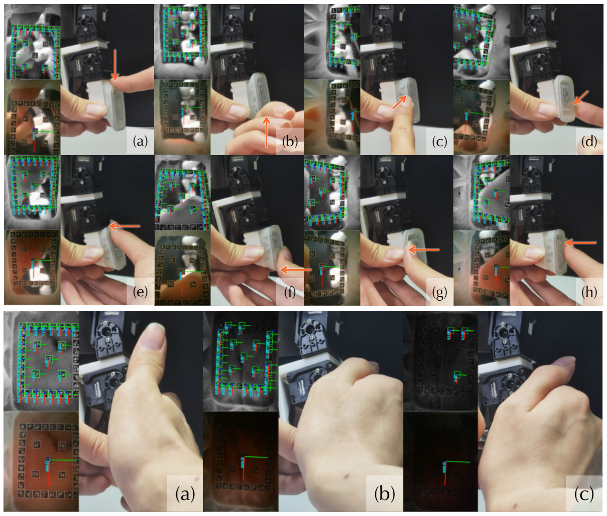


Fig. 3: **Qualitative robustness of AprilTag-based pose estimation.** Top: stable detection under force perturbations. Bottom: stable detection under lighting variation with CLAHE.

**Continuous sensing vs. explicit contact-map reconstruction.** FingerEye does not explicitly reconstruct precise contact shape or contact location. Instead, the design prioritizes continuous pre-contact perception, deformation-based post-contact feedback, and multi-view visual context for manipulation control.

### D. Deformation Estimation

The AprilTag layout is rigidly attached to the acrylic plate, so estimating its pose provides a direct proxy for deformation of the compliant ring. For each detected tag, we extract four corners and aggregate all visible corners into one multi-tag correspondence set

$$\{(\mathbf{X}_{i,j}^{\text{ref}}, \mathbf{u}_{i,j})\},$$

where $\mathbf{X}_{i,j}^{\text{ref}}$ is the 3D corner in the reference-tag frame and $\mathbf{u}_{i,j}$ is the image corner. We then estimate the plate pose by solving

$$\mathbf{T}_{\text{cam}\leftarrow\text{ref}} = \arg\min_{\mathbf{T}\in SE(3)} \sum_{i,j} \left\| \mathbf{u}_{i,j} - \pi\left(\mathbf{T}\mathbf{X}_{i,j}^{\text{ref}}\right) \right\|_2^2,$$

using EPnP followed by Levenberg–Marquardt refinement.

Compared with single-marker estimation, this multi-tag formulation provides stronger geometric constraints and reduces pose jitter under partial occlusion and local deformation. We also tested keyline-style circular markers with blob-based tracking [8]; in our setting it was less reliable under larger inter-frame deformation and illumination variation. We therefore adopt AprilTags throughout; qualitative robustness evidence is shown in Sec. IV-A, and a qualitative keyline-vs-AprilTag comparison is provided in Appendix Sec. A.1 (Fig. 16).

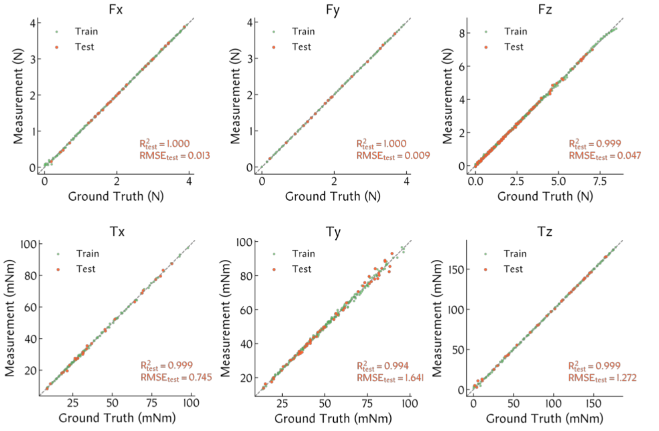


Fig. 4: **Evaluation of the force–deformation mapping of the FingerEye sensor.** Predicted wrench values from ring deformation are compared with ground truth for all six components. Green and orange points denote training and test samples, and the dashed line indicates the identity mapping. High $R^2_{\text{test}}$ and low $RMSE_{\text{test}}$ across axes confirm a strong and deterministic deformation–wrench relationship.

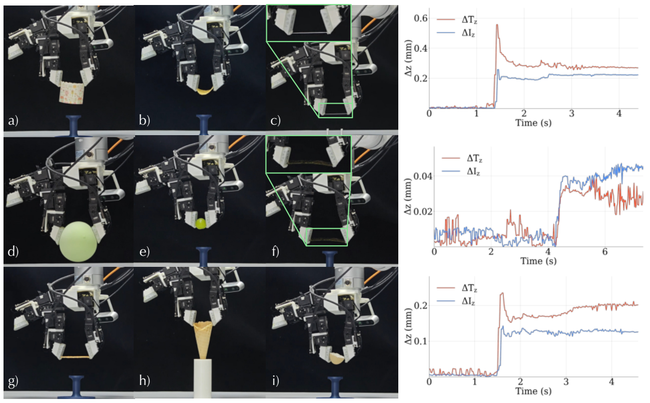


Fig. 5: **Delicate Grasping Experiments.** *Left:* representative frames of delicate grasping with the Leap Hand on nine fragile or deformable objects: (a) paper cup, (b) chip, (c) pencil lead, (d) balloon, (e) grape, (f) seaweed, (g) wafer biscuit, (h) cone, and (i) eggshell. *Right:* temporal evolution of fingertip normal deformation ($\Delta z$) for cases (c), (f), and (i).

## IV. FingerEye Sensor Experiments

In this section, we evaluate the sensing capabilities of FingerEye through a set of controlled hardware experiments. Our analysis is organized around the following guiding questions:

- Q1: Is the AprilTag-based pose estimation qualitatively robust under force and lighting variations?
- Q2: Is the vision-based deformation measurement sensitive enough to detect small and subtle wrenches?
- Q3: Does FingerEye provide a high-quality deformation signal that correlates reliably with the applied wrench?
- Q4: Can FingerEye support fine-grained and delicate grasping behaviors that require precise perception of contact onset?

### A. Qualitative Robustness of Pose Estimation

For Q1, we provide a qualitative robustness check of AprilTag-based pose estimation under force and lighting perturbations. As shown in Fig. 3 and Fig. 3, the method remains stable under both non-uniform contact deformation and low-illumination conditions. This behavior is consistent with the estimation formulation in Sec. III-D: multi-tag, multi-corner constraints improve conditioning of PnP, while CLAHE improves corner detection robustness under weak contrast. Compared with keyline-marker pipelines [8], we observed lower robustness in our setup when deformation and illumination changed simultaneously. We therefore use AprilTags in all experiments.

### B. Sensitivity Analysis

Sensitivity is a key metric for force–torque sensing. Following prior work [29, 30], we perform a fiducial-based analysis to estimate the minimum detectable forces and torques of FingerEye. The resulting minimum detectable wrench is

$$\mathbf{F}_{\min} = [4.30, 4.22, 9.93, 0.32, 0.13, 8.55]^T \ (\text{mN}, \text{mN} \cdot \text{m}).$$

These results indicate that FingerEye can detect small contact forces and subtle torques, supporting precise manipulation of delicate objects. Detailed derivations are provided in Appendix Sec. B.2.

### C. Correlation Between Wrench and Deformation

We define $g(\cdot)$ as the mapping from the six-dimensional pose change of the tag layout, $\Delta L = [\delta l_x, \delta l_y, \delta l_z, \delta \theta_x, \delta \theta_y, \delta \theta_z]^T \in \mathbb{R}^6$, to the corresponding six-dimensional force and torque, $\mathbf{F} = [f_x, f_y, f_z, t_x, t_y, t_z]^T \in \mathbb{R}^6$, such that $\mathbf{F} = g(\Delta L)$. Here, $\Delta L$ denotes the change in the estimated AprilTag pose of the acrylic plate relative to a canonical no-contact reference pose, with rotations represented as Euler angles. To quantify this mapping, we use the controlled setup in Appendix Sec. B.1, where an electronic scale measures applied forces or torques along a single axis while the acrylic plate pose is simultaneously estimated via AprilTag-based tracking. We collect over 1,000 synchronized wrench–pose pairs $(\mathbf{F}, \Delta L)$ spanning all six dimensions. For each axis, 80% of the data are used to fit a one-dimensional pose-to-force model, and the remaining 20% are held out for testing. Generalization is evaluated using the coefficient of determination ($R^2_{\text{test}}$). As shown in Fig. 4, the results validate AprilTag-based pose changes as a reliable proxy for force and torque sensing.

### D. Delicate Grasping

To evaluate the sensor's responsiveness in contact-rich manipulation, we present a qualitative study of delicate grasping, where immediate contact detection is required to avoid object damage. A Leap Hand [31] equipped with FingerEye grasps nine fragile or deformable objects (Fig. 5).

During each trial, fingers are driven at a fixed joint velocity and halted upon contact detection based on fingertip normal deformation ($\Delta z$) estimated from the AprilTag pose. Detailed thresholds, heuristics, and experimental parameters are provided in Appendix Sec. B.3. Across all objects, the sensor

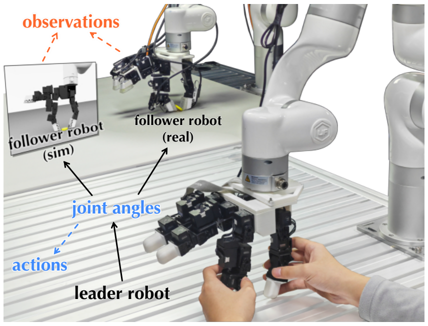


Fig. 6: **Data Collection Interface.** A human operator guides the leader robot, streaming joint positions to the follower as position target in real time.

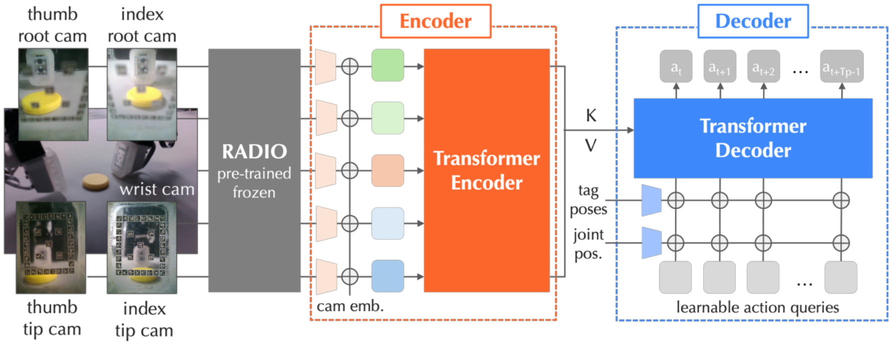


Fig. 7: **Policy Architecture.** A transformer-based policy with multi-view vision–tactile fusion and action chunking. Images from FingerEye sensors and a wrist camera are fused by a transformer encoder. A transformer decoder conditions action queries on robot state and recent tag pose history to predict action chunks.

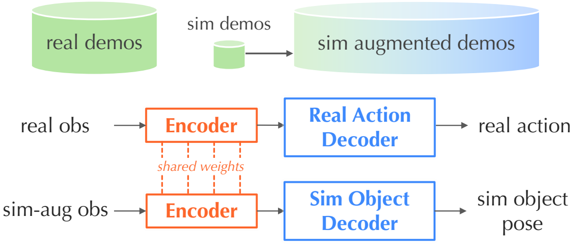


Fig. 8: **Simulation-Augmented Representation Learning.**

consistently detects contact onset and enables damage-free lifting, demonstrating the practical utility of FingerEye for delicate, contact-sensitive manipulation.

## V. Learning Dexterous Manipulation Skills with FingerEye

Having established the design and validation of FingerEye, we study how its continuous vision-tactile feedback can be exploited to learn dexterous manipulation skills from limited real-world data. To this end, we develop a learning pipeline with three components: (i) a unified data collection platform supporting real-world teleoperation and a simulation digital twin, (ii) FingerEye Policy, which fuses multi-sensor vision–tactile signals to efficiently learn complex behaviors from demonstrations, and (iii) simulation-based visual augmentation to improve robustness and generalization without additional real-world data.

### *A. Platform & Data Collection Interface*

We collect data and evaluate our policy on a fixed-base uFactory xArm7 robot equipped with a LEAP Hand (Fig. 6). Each fingertip is instrumented with a FingerEye module, and a wrist-mounted RealSense D435 provides global scene context alongside local fingertip feedback.

Demonstrations are collected using a leader–follower teleoperation setup with identical kinematics. The leader operates in passive mode with gravity compensation, while the follower tracks the streamed joint positions via PD control. Actions and observations shown in Fig. 6 are recorded at 10 Hz.

### *B. FingerEye Policy Design*

We employ a transformer-based imitation learning policy with action chunking [32, 33] (Fig. 7), which maps multimodal observations to sequences of future joint-space actions for temporally consistent control in contact-rich manipulation. The detailed policy architecture is provided in Appendix Sec. C.1 (Part B.2).

*1) Observation and Action Space:* At each time step, the policy conditions on the most recent observation $\mathbf{O}_t$ and predicts a sequence of $T_p$ future actions, of which the first $T_a$ are executed. We use $T_o = 1$, $T_p = 16$, and $T_a = 8$ in all experiments. Each observation $\mathbf{O}_t = \{\mathbf{I}_t, \mathbf{J}_t, \mathbf{T}_t\}$ consists of RGB images $\mathbf{I}_t$ from fingertip-mounted FingerEye modules and a wrist-mounted camera, the robot joint angles $\mathbf{J}_t$, and the Apriltag poses $\mathbf{T}_t$ estimated from the FingerEye tip cameras. The action sequence $\mathbf{A}_t = \{a_t, a_{t+1}, \ldots, a_{t+T_p-1}\}$ comprises joint position commands for both the hand and arm.

*2) Policy Architecture:* RGB images from all cameras (fingertip-mounted FingerEye modules and a wrist-mounted camera) are resized to $192 \times 256$ and encoded using a frozen pretrained vision backbone (RADIO [34]). The resulting features are projected to the transformer dimension and augmented with learnable per-camera embeddings (one embedding vector for each camera instance) to preserve camera identity. All visual tokens are then concatenated and processed by a transformer encoder [35] to fuse multi-view information and extract task-relevant representations.

The transformer decoder predicts an action chunk using a set of learnable action queries. To ground action generation in the current robot and contact state, we augment each action query with projected robot joint angles and a history of FingerEye tag poses from the last five timesteps before passing it to the decoder. Cross-attention [35] between the action queries and encoded observation tokens then produces temporally coherent action sequences. Detailed implementation and tensor-level specification are provided in Appendix Sec. C.1.

*3) Efficient Multi-View Training via Offline Visual Feature Caching:* To reduce multi-view training overhead, we preprocess the dataset offline by forwarding each image through

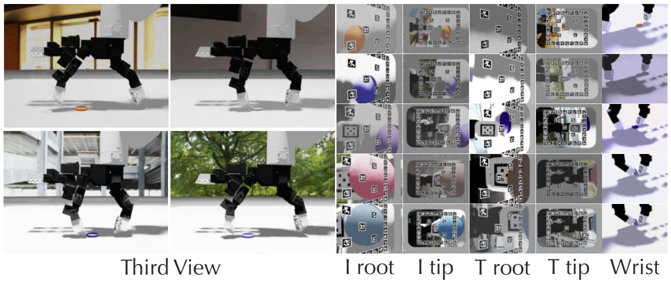


Fig. 9: Visualization of simulation visual augmentation used in our digital twin, including randomized appearance and lighting conditions.

the frozen RADIO backbone once and caching the resulting summary vectors. During policy training, raw images are replaced with cached visual features, which reduces I/O and avoids repeated backbone inference while preserving the same training objective. This optimization is particularly beneficial when using many camera views and multi-camera setups.

### C. Digital Twin and Sim-Augmented Representation Learning

Collecting large-scale real-world demonstrations for contact-rich dexterous manipulation is costly. To mitigate this and support future sim-to-real learning, we develop a simulation digital twin of FingerEye in Isaac Lab [36] (Fig. 1). In this work, we use the digital twin as a data-augmentation tool to improve generalization.

Specifically, we combine a real-world demonstration dataset with a substantially smaller set of simulated demonstrations (approximately 5× fewer) to improve robustness to object appearance variations. The digital twin enables extensive visual augmentation—randomizing image quality, illumination, and scene and object appearance—that would be difficult to achieve reliably in the real world, transforming each simulated demonstration into many visually distinct samples.

To leverage simulated data despite visual and dynamics gaps, we adopt a shared-encoder, multi-decoder training strategy (Fig. 8). Both real and simulated images are processed by a common encoder, followed by an action decoder for real data and an object-state decoder for simulated data. This encourages the encoder to capture task-relevant visual features while suppressing appearance-specific nuisances, leading to improved robustness and generalization. Comparisons with real-only and naive sim-real co-training baselines are provided in Sec. VI-D. The augmentation visualization is shown in Fig. 9. Detailed decoder architecture and training algorithm are provided in Appendix Sec. C.2 and Alg. 1.

## VI. FingerEye Policy Experiments

We evaluate **FingerEye Policy** to answer the following questions.

- **Q1: Contact sensing benefit.** Does incorporating continuous and unified vision-tactile signals from FingerEye improve policy performance over vision-only observations in contact-rich manipulation tasks?
- **Q2: Binocular sensing.** Does the binocular camera design of FingerEye provide measurable advantages over a monocular configuration for contact-rich manipulation?
- **Q3: Policy architecture.** Under the same sensing input and dataset, does our architecture provide better efficiency-performance trade-offs than ACT-style, diffusion, and pose-embedding variants?
- **Q4: Simulation-augmented representation learning.** Does simulation-augmented representation learning improve robustness to unseen object appearances compared to real-only and naive sim-real co-training?

Unless otherwise stated, all experiments are conducted on real hardware using the same dataset and evaluation metrics.

### A. Task Specification

To evaluate the usefulness of FingerEye in imitation learning, we design four manipulation tasks: *Coin Standing*, *Chip Picking*, *Letter Retrieving*, and *Syringe Manipulation*. These tasks cover scenarios requiring pre-contact perception, precise at-contact alignment, and careful post-contact force regulation. **Chip Picking (Rigid, Delicate).** A thin potato chip is randomly placed within the robot workspace with varying positions and orientations. The robot must approach the chip, establish coordinated multi-finger contact, and lift it without breaking or slipping. The task succeeds if the chip is lifted off the table and held stably. This task evaluates fine-grained contact localization and force modulation on small, fragile rigid objects. Training and test configurations are shown in Fig. 11. **Coin Standing (Rigid, Contact-Sensitive).** A coin initially lies flat on the table at a random pose. The robot must select appropriate contact points to push, wedge, and reorient the coin, stand it upright, and disengage without toppling it. The task is successful if the coin remains upright after the robot disengages. This task requires precise contact initiation, continuous contact monitoring, and delicate force regulation on rigid objects with unstable equilibria. **Letter Retrieving (Deformable, Thin-Shell, Long-Horizon).** An envelope with a letter is tilted and partially suspended in the workspace. The robot must approach the edge, open the flap, wedge the gap, pinch the letter, and extract it without tearing or collapsing the envelope. The task succeeds when the letter is fully retrieved and evaluates perception-driven contact sequencing and force control on thin, deformable objects. **Syringe Manipulation (Articulated, Functional).** A medical syringe is randomly positioned within the workspace. The robot must reorient the syringe, establish a stable multi-finger grasp, transport it to a container, and press the plunger to expel the liquid. The task is successful if all liquid is injected into the container. This long-horizon task requires coordinated multi-contact control and precise force application on articulated objects.

### B. Comparisons

We compare methods from two complementary perspectives. **Sensing modalities (real world and simulation).** For modality comparisons in both real-world tasks (Fig. 12) and simulation coin standing (Fig. 13, left), we evaluate:

- **Wrist-Camera Only.** The policy receives only RGB frames from the wrist-mounted camera. This setting represents the most common vision-only baseline.

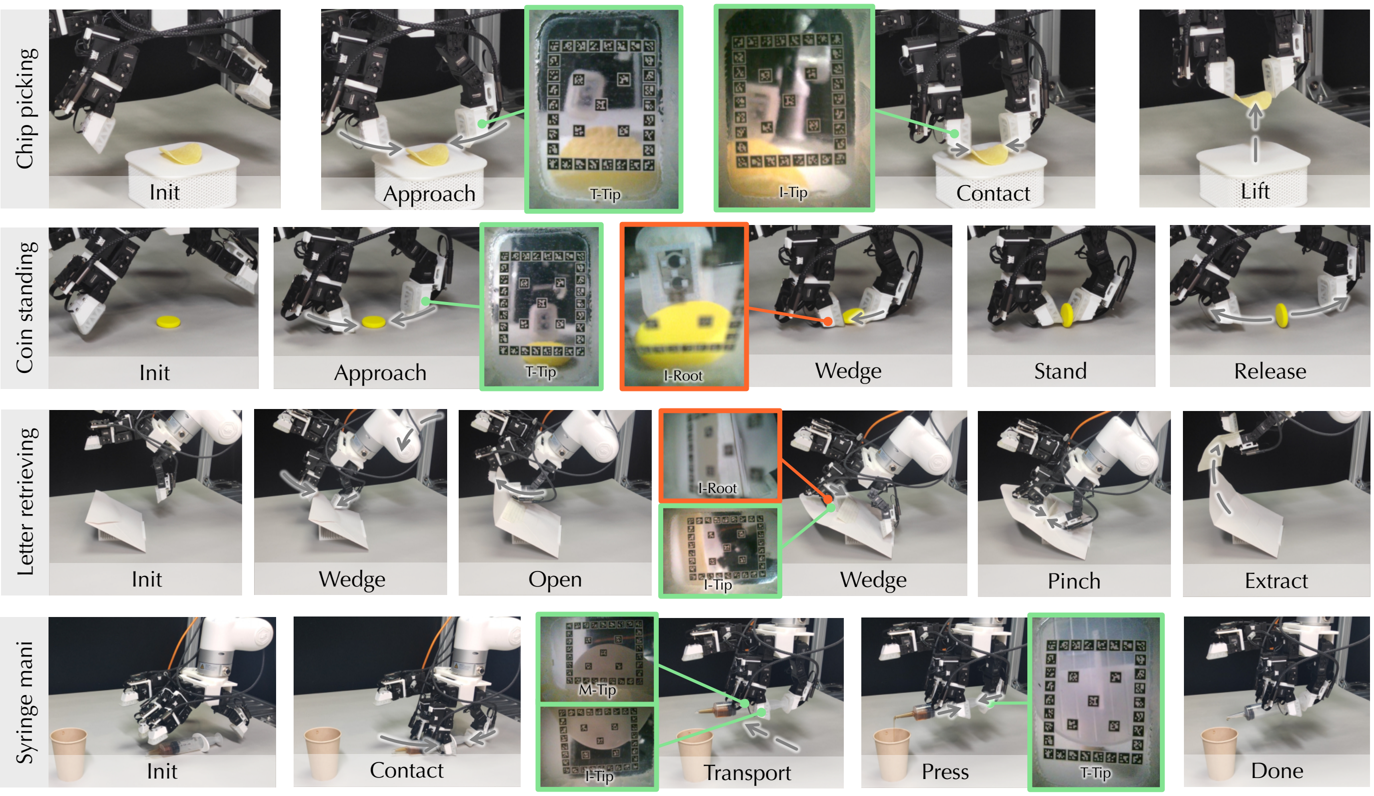


Fig. 10: **Real-world experimental rollouts.** We evaluate the proposed vision–tactile sensing framework on four representative tasks spanning rigid, deformable, and articulated objects.

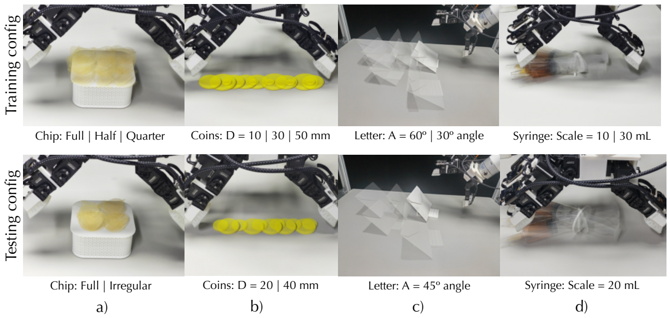


Fig. 11: **Real-world task configurations for training and testing.**

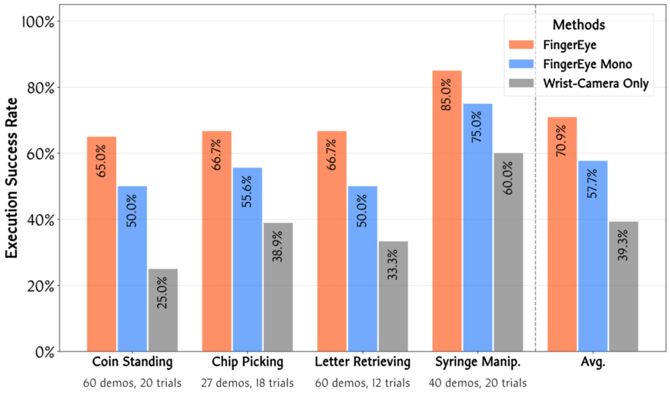


Fig. 12: **Quantitative results across tasks and sensing modalities.**

- **FingerEye Mono.** In addition to the wrist camera, each FingerEye contributes a single monocular RGB stream.
- **FingerEye.** In addition to the wrist camera, each FingerEye module provides binocular RGB frames.

**Policy architectures (simulation).** For architecture comparisons (Fig. 13, right), we fix the sensing modality (FingerEye) and training dataset, and compare:

- **ACT ResNet Encoder**, replacing RADIO summary features and per-camera learnable embeddings with dense tokens from a trainable ResNet-18 plus 2D sinusoidal positional embeddings, following ACT-style design [37, 38, 33].
- **Diffusion Policies**, using a UNet-based diffusion policy with short (20-step) and long (100-step) denoising schedules [32].
- **RoboPanoptes Pose Emb.**, replacing learnable per-camera embeddings with explicit camera-pose embeddings [39].
- **FingerEye Policy (Ours)**, using RADIO summary features and learnable per-camera embeddings.

These comparisons evaluate both sensing and architecture under matched datasets and protocols.

### C. Experimental Analysis and Key Takeaways

We analyze results from sensing modalities and policy architecture. Using success rates (Fig. 12), simulation results (Fig. 13), and representative failure cases (Fig. 14), we draw the following conclusions.

**Modalities: local contact sensing substantially improves reliability.** Across all tasks, policies using FingerEye or FingerEye Mono significantly outperform the wrist-camera-

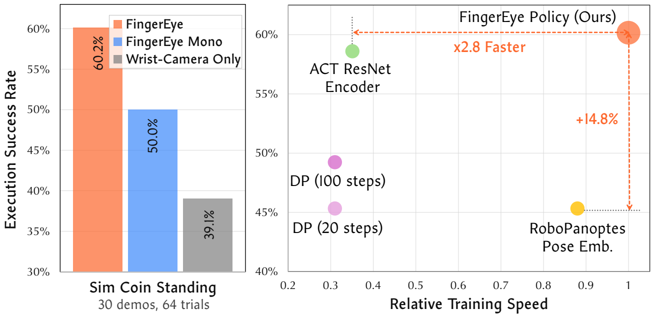


Fig. 13: **Simulation results on coin standing.** *Left*: execution success rates under different sensing modalities. *Right*: training speed and final success across policy architectures under identical FingerEye visual inputs and training data. Relative training speed is normalized to FingerEye Policy.

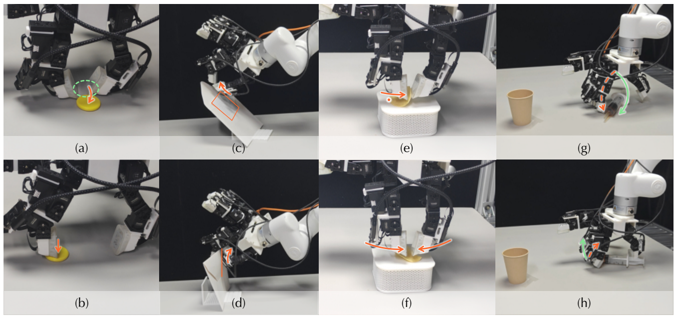


Fig. 14: **Representative failure cases of baseline policies.** (a) Slight contact point offset pushes the coin away instead of wedging it. (b) Imprecise visual localization causes the finger to miss the coin. (c) Incorrect inference of a successful pinch leads to lifting without grasping the letter. (d) Failure to detect the envelope edge prevents flap opening. (e,f) Missed chip edges result in unstable contact and dropping during lifting. (g) Wrist-camera-only policy fails to reorient the syringe and strikes it away. (h) Monocular fingertip policy fails to maneuver around and engage the syringe edge, contacting only one side and pushing it out of range.

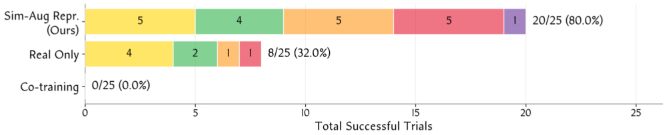


Fig. 15: Comparison of execution success rates across five coin-color variants (Yellow, Green, Orange, Red, and Purple). Each row represents a different training methodology, with segments indicating the number of successful trials (out of 5) for each specific color.

only baseline, *with FingerEye achieving over* 30% *average improvement*. This indicates that global vision alone does not provide sufficient information for reliable contact-rich manipulation. In many failures of the wrist-only baseline, the robot reaches the vicinity of the object but with slight errors in contact point or approach direction. These small inaccuracies are sufficient to cause failure, such as pushing the object away, sliding past thin edges, or destabilizing the interaction, as shown in Fig. 14. We hypothesize that this behavior arises because wrist-mounted vision provides only coarse, distal cues and cannot reliably resolve fine-scale object surfaces or infer local interaction forces at close range. These errors arise because the contact state must be inferred indirectly from distal visual cues, which become unreliable under occlusion and viewpoint changes. By contrast, FingerEye provides continuous fingertip-level vision-tactile feedback that directly reflects contact onset and evolution, enabling the policy to adjust its actions as interaction unfolds.

The same ordering also appears in simulation (Fig. 13, left): FingerEye > FingerEye Mono > Wrist-Camera Only. Although the absolute gap between FingerEye and wrist-only is smaller in simulation, the main modality-dependent failures still concentrate in pre-contact alignment and wedging, while final balancing is relatively easier in simulation due to cleaner sensing and contact dynamics.

**Modalities: binocular sensing outperforms monocular.** Monocular fingertip sensing performs better than wrist-mounted vision but is consistently worse than FingerEye. While local fingertip images help the policy reach the object, monocular sensing often fails to place the fingertip at the correct distance for contact. As a result, the finger may stop short, slide past object edges, or hit the object from an unfavorable direction. For example, during syringe manipulation, the monocular policy sometimes does not move the finger far enough around the syringe during reorientation, instead contacting one side and pushing it away (Fig. 14(h)). Similarly, in chip picking, the fingertip may slide over the chip edge without establishing contact, leading to missed grasps or object drop (Fig. 14(e)). Binocular fingertip sensing provides more reliable distance cues at close range, enabling more accurate fingertip placement and more stable contact.

**Architecture: our design is faster and stronger under matched inputs.** Under identical FingerEye visual inputs and demonstrations, FingerEye Policy reaches comparable or better final performance with about 2.8× **faster relative training speed** and +14.8% **higher execution success** than the strongest baselines (Fig. 13, right). Compared with ACT ResNet Encoder, pretrained ViT summary tokens reduce per-camera token count and improve multi-view efficiency. Diffusion policies become competitive only with long denoising schedules, but incur substantially slower optimization and inference; direct action regression with action chunking avoids iterative denoising while keeping strong performance [40]. Replacing explicit pose embeddings with learnable per-camera embeddings also improves results, suggesting that lightweight camera-indexed embeddings provide sufficient view identity cues for contact-rich manipulation.

### D. Sim-Augmented Representation Learning Experiments

We evaluate simulation-augmented representation learning in a controlled appearance generalization setting using the *Coin Standing* task. In this experiment, all real-world and simulated demonstrations are collected using a *yellow* coin. We evaluate on five coin colors (yellow, green, orange, red, and purple). Following Sec. V-C, both real and simulated images are processed by a shared encoder, with an action decoder for real data and an object-state decoder for simulated data. This setting isolates the effect of visual appearance shift while keeping geometry and dynamics fixed.

*a) Comparisons:* We compare three training strategies under the same policy architecture, dataset and evaluation protocol:

- **Real-Only.** 60 real-world demonstrations collected with the yellow coin.
- **Sim&Real Co-Training.** 50 real-world demonstrations combined with 10 simulated demonstrations. The simulated demonstrations are visually augmented to produce 320 samples and are trained jointly with real data using action supervision.
- **Sim-Augmented Representation Learning (Ours).** 50 real-world demonstrations combined with 10 simulated demonstrations, augmented to 320 samples. Method is as described in Sec. V-C and Fig. 8.

*b) Results & Analysis:* Fig. 15 reports the number of successful trials (out of five) for each coin color. The real-only baseline performs reasonably on the training color (yellow) but generalizes poorly to unseen colors. Naive sim2real co-training fails across all colors despite extensive visual augmentation, indicating that directly mixing simulated and real action supervision is ineffective under dynamics mismatch. In contrast, simulation-augmented representation learning generalizes well to most unseen colors, achieving high success rates on yellow, green, orange, and red coins, with partial success on the purple coin. Our interpretation is that the purple coin is harder because it introduces a larger visual shift relative to the default training appearance (yellow coin), and therefore likely requires more training data to generalize as reliably as the other colors. These results suggest that, for contact-rich dexterous manipulation, the sim2real gap is dominated by interaction dynamics rather than visual appearance. Using simulation for auxiliary visual supervision—while learning actions only from real data—allows augmented simulation images to improve appearance robustness without introducing erroneous action labels. Detailed architecture and training algorithm are provided in Appendix Sec. C.2 and Alg. 1.

## VII. Limitations and Conclusions

While FingerEye demonstrates strong performance on contact-rich dexterous manipulation tasks, several limitations suggest promising directions for future work. The current camera placements and tag layouts are empirically designed; learning or optimizing sensor geometry could further improve coverage and robustness across diverse contact configurations. In addition, our system currently focuses on single-arm, fixed-base manipulation, and extending FingerEye to mobile, bimanual, or humanoid platforms would enable richer and more realistic interaction scenarios. Looking ahead, the simulation introduced in this work provides a foundation for scalable data generation. Such scalable data can support reinforcement learning, hybrid imitation-reinforcement learning, and recovery behaviors beyond demonstrated trajectories, and more broadly enable the training and adaptation of foundation models for contact-rich dexterous manipulation. Overall, we view FingerEye as a step toward continuous and unified vision-tactile sensing for dexterous manipulation. We hope this work will inspire future research on integrated sensing, scalable data generation, and foundation-model-based learning for safe and general-purpose dexterous robotic interaction in real-world environments.

## Acknowledgement

We thank Junyuan Cui for assistance with the simulation setup; Jinxuan Zhu, Jingxiang Guo, Zixuan Liu, Chongkai Gao, Chenrui Tie, Quantao Li, Yiwen Hou, and Junting Chen for their valuable discussions; and Hengxu Yan, Zhijie Zhang, and Viking Wang for assistance with the hardware setup. This work was supported in part by the President's Graduate Fellowship and RoboScience. The views and conclusions expressed herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the sponsors.

## References

[1] Matthew T Mason. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):1–28, 2018.

[2] Tailai Cheng, Kejia Chen, Lingyun Chen, Liding Zhang, Yue Zhang, Yao Ling, Mahdi Hamad, Zhenshan Bing, Fan Wu, Karan Sharma, et al. Tacumi: A multi-modal universal manipulation interface for contact-rich tasks. *arXiv preprint arXiv:2601.14550*, 2026.

[3] Fang Wan and Chaoyang Song. Seethrufinger: See and grasp anything with a soft touch. *arXiv preprint arXiv:2312.09822*, 2023.

[4] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[5] Changyi Lin, Han Zhang, Jikai Xu, Lei Wu, and Huazhe Xu. 9dtact: A compact vision-based tactile sensor for accurate 3d shape reconstruction and generalizable 6d force estimation. *IEEE Robotics and Automation Letters*, 2023.

[6] Atsushi Yamaguchi and Christopher G. Atkeson. Implementing tactile behaviors using fingervision. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 241–248. IEEE, 2017. doi: 10.1109/HUMANOIDS.2017.8246891.

[7] Sheeraz Athar, Gaurav Patel, Zhengtong Xu, Qiang Qiu, and Yu She. Vistac toward a unified multimodal sensing finger for robotic manipulation. *IEEE Sensors Journal*, 23(20):25440–25450, 2023. doi: 10.1109/JSEN.2023.3310918.

[8] Yuyang Li, Yinghan Chen, Zihang Zhao, Puhao Li, Tengyu Liu, Siyuan Huang, and Yixin Zhu. Simultaneous tactile-visual perception for learning multimodal robot manipulation. *arXiv preprint arXiv:2512.09851*, 2025.

[9] Patrick Lancaster, Pratik Gyawali, Christoforos Mavrogiannis, Siddhartha S Srinivasa, and Joshua R Smith. Optical proximity sensing for pose estimation during in-hand manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11818–11825. IEEE, 2022.

[10] Quan Khanh Luu, Dinh Quang Nguyen, Nhan Huu Nguyen, Nam Phuong Dam, and Van Anh Ho. Vision-based proximity and tactile sensing for robot arms: Design, perception, and control. *IEEE Transactions on Robotics*, 2025.

[11] Francois R Hogan, Jean-François Tremblay, Bobak H Baghi, Michael Jenkin, Kaleem Siddiqi, and Gregory Dudek. Finger-sts: Combined proximity and tactile sensing for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(4):10865–10872, 2022.

[12] Etienne Roberge, Guillaume Fornes, and Jean-Philippe Roberge. Stereotac: A novel visuotactile sensor that combines tactile

sensing with 3d vision. *IEEE Robotics and Automation Letters*, 8(10):6291–6298, 2023.

[13] Trevor Ablett, Oliver Limoyo, Adam Sigal, Affan Jilani, Jonathan Kelly, Kaleem Siddiqi, Francois Hogan, and Gregory Dudek. Multimodal and force-matched imitation learning with a see-through visuotactile sensor. *IEEE Transactions on Robotics*, 41: 946–959, 2025. doi: 10.1109/TRO.2024.3521864.

[14] Jikai Xu, Lei Wu, Changyi Lin, Ding Zhao, and Huazhe Xu. Dtactive: A vision-based tactile sensor with active surface. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21664–21670. IEEE, 2025.

[15] Yueshi Dong, Jieji Ren, Zhenle Liu, Zhanxuan Peng, Zihao Yuan, Ningbin Zhang, and Guoying Gu. Look-to-touch: A vision-enhanced proximity and tactile sensor for distance and geometry perception in robotic manipulation. *IEEE/ASME Transactions on Mechatronics*, 2026.

[16] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics*, 5(2):216–227, 2018.

[17] Shixin Zhang, Zixi Chen, Yuan Gao, Weiwei Wan, Jianhua Shan, Hongxiang Xue, Fuchun Sun, Yiyong Yang, and Bin Fang. Hardware technology of vision-based tactile sensor: A review. *IEEE Sensors Journal*, 22(22):21410–21427, 2022.

[18] Haoran Li, Yijiong Lin, Chenghua Lu, Max Yang, Efi Psomopoulou, and Nathan F. Lepora. Classification of vision-based tactile sensors: A review. *IEEE Sensors Journal*, 25(19):35672–35686, 2025. doi: 10.1109/JSEN.2025.3599236.

[19] Yi-Hang Xin, Kai-Ming Hu, Rui-Jia Xiang, Yu-Ling Gao, Jun-Feng Zhou, Guang Meng, and Wen-Ming Zhang. Vision-based tactile sensing: From performance parameters to device design. *Applied Physics Reviews*, 12(2), 2025.

[20] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018.

[21] Won Kyung Do and Monroe Kennedy. Densetact: Optical tactile sensor for dense shape reconstruction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6188–6194. IEEE, 2022.

[22] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasy, T. Ikeda, and R. Tedrake. Soft-bubble grippers for robust and perceptive manipulation. *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[23] Branden Romero, Hao-Shu Fang, Pulkit Agrawal, and Edward Adelson. Eyesight hand: Design of a fully-actuated dexterous robot hand with integrated vision-based tactile sensors and compliant actuation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1853–1860. IEEE, 2024.

[24] Jialiang Zhao, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Edward Adelson. Polytouch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. *arXiv preprint arXiv:2504.19341*, 2025.

[25] Son Bui, Duy Le, Tu Nguyen, Son Nguyen, Son Tran, Luc Tran, and Thong Pham. Digiteye: A transparent soft tactile sensor for robust multi-modal perception. *Journal of Machine Engineering*, 25:91–105, December 2025. doi: 10.36897/jme/213851.

[26] Francois R. Hogan, Michael Jenkin, Sahand Rezaei-Shoshtari, Yogesh Girdhar, David Meger, and Gregory Dudek. Seeing through your skin: Recognizing objects with a novel visuotactile sensor. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1218–1227, January 2021.

[27] Yueshi Dong, Jieji Ren, Zhenle Liu, Zhanxuan Peng, Zihao Yuan, Ningbin Zhang, and Guoying Gu. Look-to-touch: A vision-enhanced proximity and tactile sensor for distance and geometry perception in robotic manipulation. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2504.10280. arXiv:2504.10280.

[28] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561.

[29] Rui Ouyang and Robert Howe. Low-cost fiducial-based 6-axis force-torque sensor. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1653–1659. IEEE, 2020.

[30] Jinxuan Zhu, Zihao Yan, Yangyu Xiao, Jingxiang Guo, Chenrui Tie, Xinyi Cao, Yuhang Zheng, and Lin Shao. Shapeforce: Low-cost soft robotic wrist for contact-rich manipulation. *arXiv preprint arXiv:2511.19955*, 2025.

[31] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[32] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[33] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[34] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12490–12500, 2024.

[35] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[36] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter, Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the*

*IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[38] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[39] Xiaomeng Xu, Dominik Bauer, and Shuran Song. Robopanoptes: The all-seeing robot with whole-body dexterity. *arXiv preprint arXiv:2501.05420*, 2025.

[40] Chaoyi Pan, Giri Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. *arXiv preprint arXiv:2512.01809*, 2025.

[41] Eric T. Chang, Peter Ballentine, Zhanpeng He, Do-Gon Kim, Kai Jiang, Hua-Hsuan Liang, Joaquin Palacios, William Wang, Pedro Piacenza, Ioannis Kymissis, and Matei Ciocarlie. Spikeatac: A multimodal tactile finger with taxelized dynamic sensing for dexterous manipulation, 2025. URL https://arxiv.org/abs/2510.27048.

[42] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[43] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

# Appendix

*Appendix Contents*

## A. *Additional Hardware Details*

*1) Comparison with Keyline Markers:* We compare our AprilTag-based pipeline with a keyline-marker pipeline inspired by TacThru [8]. The keyline approach detects circular markers with blob detection, then relies on nearest-neighbor association and temporal filtering. In our setup, this pipeline is less stable when deformation between frames is large and when ambient illumination varies, because both matching and blob responses become more brittle.

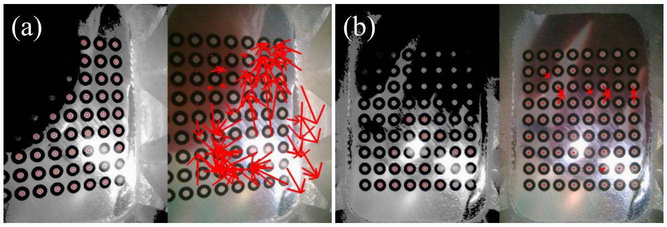


Fig. 16: Qualitative comparison between keyline-marker tracking and our AprilTag-based pipeline under challenging deformation and lighting conditions.

In contrast, our method uses explicit tag identities and multi-corner geometric constraints in a unified PnP solve, which provides better conditioning and robustness for pose estimation.

*2) Comparison with Gelsight:* We attempted to use Gelsight for the task of *coin standing*, but encountered several issues during the data collection stage of teleoperation. For instance, Gelsight was unable to detect the contact between the sensor and the table surface due to the lack of the deformation at its sides. Additionally, the square shape of Gelsight hinders its ability to interact smoothly with the coin, preventing it from being able to wedge the coin. A representative failure of gelsight is shown in Fig. 17. In contrast, FingerEye uses its wedge-shaped soft ring to detect the contact between the table and the robot, allowing it to wedge the coin and capture sufficient, useful information for manipulation.

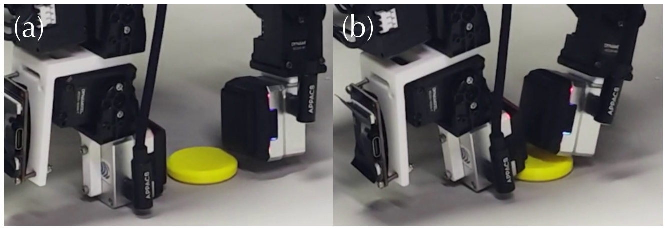
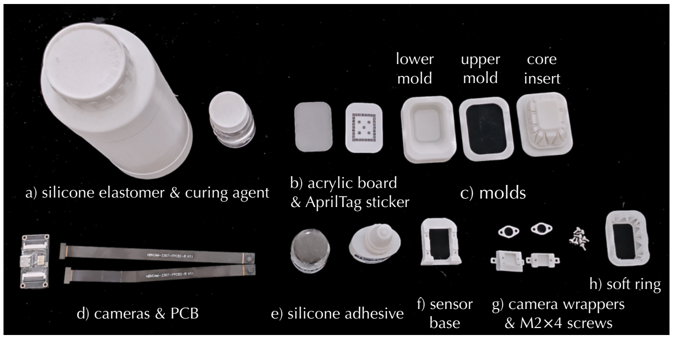


Fig. 17: Failure case of Gelsight in the *coin standing* task. (a) shows the initial approach, (b) illustrates the failure to capture the coin.

*3) Peripheral Deformation Enabled by a Compliant Ring:* The soft ring of FingerEye is mechanically bonded to and surrounds the acrylic cover, forming a compliant boundary at the fingertip periphery. In contrast to designs that enclose a deformable medium within a rigid structural ring, this compliant interface allows contact and deformation to occur not only at the central surface but also along the fingertip edge.

As a result, FingerEye can sense interactions involving lateral contact, edge engagement, and peripheral loading, which are common in dexterous manipulation tasks such as wedging, sliding, and reorientation. This expanded deformation region effectively increases the sensing coverage beyond the central contact area, providing richer interaction cues during contact-rich manipulation. Representative examples are shown in Fig. 3.



lower mold
upper mold
core insert
a) silicone elastomer & curing agent
b) acrylic board & AprilTag sticker
c) molds
d) cameras & PCB
e) silicone adhesive
f) sensor base
g) camera wrappers & M2×4 screws
h) soft ring

Fig. 18: Materials used for FingerEye fabrication, including silicone elastomer and curing agent, camera modules, acrylic plate with AprilTag sticker, silicone adhesive, and 3D-printed mold components.

*4) Fabrication Details:* Our fabrication process consists of the following steps.

1) **3D Printing.** 3D-print the sensor base, camera wrappers, and mold components, including the lower mold, upper mold, and core insert.
2) **Silicone Preparation and Casting.** Prepare the silicone elastomer by mixing the silicone base with the silicone curing agent at a mass ratio of base : curing agent $= 100 : (1 \sim 3)$. Pour the mixture into the lower mold until the cavity is filled. Assemble the upper mold and core insert

into a single unit, then invert and press the assembly onto the lower mold. Allow the silicone to cure under gravity and compression for more than 4 hours. After curing, remove the upper mold and core insert, then demold the cast part to obtain the soft silicone ring.

3) **Acrylic Plate and Tag Preparation.** Apply the AprilTag sticker to the inner surface of the acrylic plate. Bond the soft silicone ring to the AprilTag sticker using silicone adhesive (JL-401), ensuring uniform contact. Allow the adhesive to fully cure.
4) **Camera Assembly.** Secure the camera wrappers to the sensor base using M2×4 screws, and install the camera modules into the wrappers.
5) **Wiring and Connection.** Connect the assembled sensor to a host PC using a USB-C cable.

*5) Bill of Materials:* The detailed cost is listed in Table I.

TABLE I: Bill of Materials for One FingerEye Module

| Item | Cost (USD) |
|---|---|
| Acrylic plate | 0.28 |
| AprilTag sticker | 0.42 |
| Silicone elastomer & curing agent (20 g) | 0.10 |
| Silicone adhesive (JL-401, ∼0.5 ml) | 0.17 |
| Soft silicone ring | 0.02 |
| Sensor base (3D-printed) | 0.44 |
| Camera wrappers (3D-printed) | 0.45 |
| Soft-ring mold (3D-printed) | 1.78 |
| M2×4 screws (8 pcs) | 0.04 |
| Camera modules (×2) | 56.49 |
| Total | 60.19 |

### *B. Additional Hardware Experiment Details*

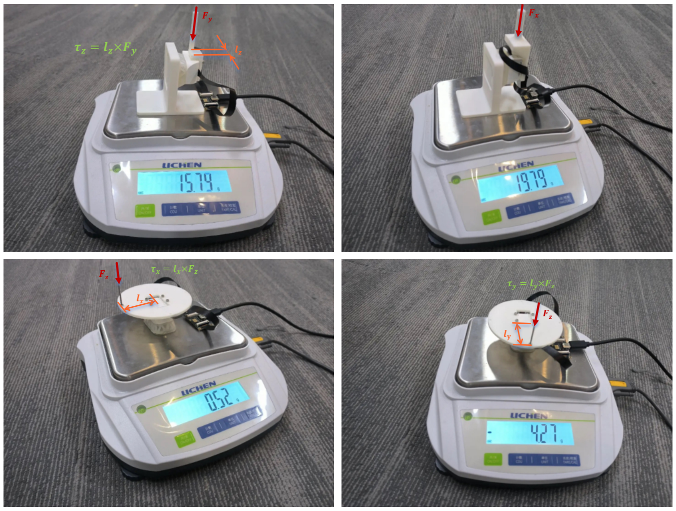


Fig. 19: Experiment setup

*1) Wrench–Deformation Correlation Experiment Setup:* To identify the mapping between the deformation of FingerEye and the applied wrench, we use the controlled hardware setup shown in Fig. 19. A digital scale is used to measure the applied forces, while a set of 3D-printed fixtures enables force application from different directions and at different contact locations on the sensor. During each trial, forces are applied along one axis at a time. For torque measurements, a force is applied at the outer edge of the fixture, and the corresponding torque is computed using the known lever arm. The pose change of the acrylic plate relative to a no-contact reference is simultaneously estimated using AprilTag-based tracking. Following this procedure, we collect over 1,000 synchronized pairs of applied wrench and measured deformation, spanning all six translational and rotational degrees of freedom. Model formulation, fitting procedure, and quantitative evaluation are described in the main paper.

*2) Sensitivity Analysis (Details):* This section provides the detailed derivation of the sensitivity analysis summarized in the main paper. Following fiducial-based sensitivity analysis in prior work [29, 30], we estimate the minimum detectable pose change of FingerEye from pixel-level localization accuracy and propagate it to force–torque sensitivity using the calibrated wrench–deformation model.

**Subpixel localization assumption.** We assume a subpixel localization accuracy of $d_R = 0.25$ px for AprilTag detection, consistent with prior work.

**Translational sensitivity.** Let $w_{\text{tag}}$ denote the physical width of the AprilTag and $w_{\text{img}}$ its observed width in the camera image. A displacement of $d_R$ pixels corresponds to a minimum detectable translation

$$\Delta l_{\min} = \frac{w_{\text{tag}}}{w_{\text{img}}} d_R. \tag{1}$$

Using $w_{\text{tag}} = 2.0$ mm and $w_{\text{img}} = 37$ px, we obtain

$$\Delta l_{\min} \approx 0.0135 \text{ mm}.$$

**Rotational sensitivity.** For a reference rotation angle $\theta$, the corresponding pixel displacement at radius $r$ (half-diagonal of the tag image patch) is given by the chord length $2r\sin(\theta/2)$. A $d_R$-pixel displacement therefore corresponds to a minimum detectable rotation.

$$\Delta\theta_{\min} = \frac{\theta}{2r\sin(\theta/2)} d_R. \tag{2}$$

Using $\theta = \pi/12$ and $r = 18.5$ px, we obtain

$$\Delta\theta_{\min} \approx 0.0136 \text{ rad}.$$

**Pose sensitivity summary.** The resulting minimum detectable 6-DoF pose change is

$$\Delta L_{\min} = [0.0135,\ 0.0135,\ 0.0135,\ 0.0136,\ 0.0136,\ 0.0136]^{\mathsf{T}}, \tag{3}$$

where the first three entries are in millimeters and the last three are in radians.

**Propagation to wrench sensitivity.** Propagating $\Delta L_{\min}$ through the calibrated linear wrench–deformation model yields the minimum detectable wrench

$$\mathbf{F}_{\min} = [4.30,\ 4.22,\ 9.93,\ 0.32,\ 0.13,\ 8.55]^{\mathsf{T}}, \tag{4}$$

where the first three entries correspond to force sensitivity (mN) and the last three correspond to torque sensitivity (N·mm). These results support the claim in the main paper that FingerEye is capable of detecting small contact forces and subtle torques, enabling precise control in contact-rich manipulation tasks.

*3) Detailed Delicate Grasping Setup and Hyper-parameters:* Inspired by prior work on contact-sensitive grasping, such as SpikeAtac [41], we design a set of delicate grasping experiments to evaluate FingerEye in scenarios requiring early contact detection and gentle interaction. These experiments aim to validate whether vision-based tactile feedback from FingerEye can reliably indicate contact onset and enable responsive stopping and lifting behaviors.

Due to its vision-based sensing modality, FingerEye operates at moderate joint velocities compatible with the camera frame rate. All grasping motions follow a predefined joint-space trajectory generated by linear interpolation between an initial pose and a target pose. Joint commands are issued at a fixed control interval of 0.02 s per frame.

To detect contact, we monitor the change in the acrylic-panel normal displacement $\Delta z$ estimated from AprilTag tracking. When $\Delta z$ exceeds a predefined threshold, the system stops the approach motion and lifts the object. This threshold serves as a simple and intuitive contact indicator that is consistently applied across experiments.

In most settings, a custom 3D-printed support platform is used to elevate objects, facilitating stable grasping of thin, small, or fragile items such as pen refills, cookies, seaweed, and ice-cream cones.

The hyperparameters used in these experiments are summarized in Table II. The contact threshold specifies the minimum detectable normal displacement for triggering contact, while the total number of frames determines the duration of the approach phase and the resulting joint velocity. These values are chosen conservatively to ensure gentle interaction rather than aggressive or high-speed grasping.

TABLE II: Hyper-parameters for delicate grasping experiments

| Object | Contact threshold (mm) | Total frames |
|---|---|---|
| Chip | 0.10 | 30 |
| Eggshell | 0.05 | 20 |
| Cone | 0.10 | 20 |
| Cookie | 0.05 | 5 |
| Balloon | 0.05 | 3 |
| Pencil | 0.02 | 15 |
| Paper | 0.01 | 50 |
| Paper cup | 0.005 | 10 |
| Grape | 0.006 | 10 |
| Seaweed | 0.01 | 150 |

*4) Full Visualization of Force Curves:* In our entire delicate experiment, the experimental setup and its corresponding curves are shown in Fig. 20.

### *C. Additional Policy Learning Details*

*1) Detailed Policy Architecture:* We describe the FingerEye policy architecture in detail, explicitly specifying all inputs, learnable components, and tensor dimensions. The same architecture is used across all experiments unless otherwise stated.

**Temporal structure.** At each control step $t$, the policy conditions on the most recent $T_o$ observations and predicts a sequence of $T_p$ future actions. In all experiments, we set $T_o = 1$, $T_p = 16$, and execute the first $T_a = 8$ actions in a receding-horizon manner.

**Observation space.** Each observation is defined as

$$\mathbf{O}_t = \{\mathbf{I}_t, \mathbf{J}_t, \mathbf{T}_t\}.$$

- **Images.** $\mathbf{I}_t \in \mathbb{R}^{N_{\text{cam}} \times 3 \times 192 \times 256}$ denotes RGB images from $N_{\text{cam}}$ cameras, including fingertip-mounted FingerEye cameras and a wrist-mounted camera.
- **Proprioception.** $\mathbf{J}_t \in \mathbb{R}^{d_s}$ represents the robot state, consisting of joint angles.
- **Tag poses (optional).** $\mathbf{T}_t \in \mathbb{R}^{n_{\text{tag\_steps}} \times n_{\text{tag}} \times 9}$ stores Apriltag poses over the past $n_{\text{tag\_steps}} = 5$ timesteps. Each pose is parameterized by a 3D position and a 6D continuous rotation representation given by the first two columns of the rotation matrix [42].

**Visual encoding with foundation-model summaries.** Let $i \in \{1, \ldots, N_{\text{cam}}\}$ index the camera instances. Each image $\mathbf{I}_t^{(i)}$ is processed independently by a frozen RADIO [34] vision backbone. For each camera, the backbone outputs a global summary vector

$$\mathbf{s}_t^{(i)} \in \mathbb{R}^{2304},$$

which aggregates semantic and geometric information from the full image. For $T_o = 1$, stacking across cameras yields a matrix

$$\mathbf{S}_t = \left[\mathbf{s}_t^{(1)}, \ldots, \mathbf{s}_t^{(N_{\text{cam}})}\right]^\top \in \mathbb{R}^{N_{\text{cam}} \times 2304}.$$

In contrast to dense feature grids, we rely exclusively on these summary tokens, which provide a compact yet expressive representation that enables more efficient training with multi-view cameras while maintaining comparable performance in our experiments.

**Per-camera projection and learnable embeddings.** Each camera has an independent projection network

$$\phi^{(i)} : \mathbb{R}^{2304} \to \mathbb{R}^{512},$$

implemented as a two-layer MLP with GELU activations. This produces projected visual tokens

$$\mathbf{v}_t^{(i)} = \phi^{(i)}(\mathbf{s}_t^{(i)}) \in \mathbb{R}^{512}.$$

To preserve camera identity, we associate each camera with a learnable embedding

$$\mathbf{e}^{(i)} \in \mathbb{R}^{512},$$

and form the final camera tokens

$$\tilde{\mathbf{v}}_t^{(i)} = \mathbf{v}_t^{(i)} + \mathbf{e}^{(i)}.$$

This representation allows the policy to reason over heterogeneous viewpoints using a simple and explicit camera-indexed tokenization.

**Multi-view transformer encoder.** The set of camera tokens

$$\tilde{\mathbf{V}}_t = \{\tilde{\mathbf{v}}_t^{(1)}, \ldots, \tilde{\mathbf{v}}_t^{(N_{\text{cam}})}\} \in \mathbb{R}^{N_{\text{cam}} \times 512}$$

is processed by a transformer encoder with 4 layers, 8 attention heads, and feedforward dimension 2048. The encoder outputs fused visual features

$$\mathbf{Z}_t \in \mathbb{R}^{N_{\text{cam}} \times 512},$$

which serve as the conditioning context for action generation.

**State and tag conditioning.** Robot state vectors are projected via

$$\psi_s : \mathbb{R}^{T_o \cdot d_s} \to \mathbb{R}^{512},$$

yielding a state embedding $\mathbf{s}_t^{\text{state}} \in \mathbb{R}^{512}$. When available, tag pose histories are flattened to

$$\mathbf{T}_t^{\text{flat}} \in \mathbb{R}^{n_{\text{tag\_steps}} \cdot n_{\text{tag}} \cdot 9}$$

and encoded using a low-rank FiLM-style [43] module to produce a modulation vector

$$\gamma_t \in \mathbb{R}^{512}.$$

**Action query initialization and decoding.** We maintain $T_p$ learnable action query embeddings

$$\mathbf{q}_k \in \mathbb{R}^{512}, \quad k = 1, \ldots, T_p.$$

Each query is conditioned as

$$\tilde{\mathbf{q}}_k = (\mathbf{q}_k + \mathbf{s}_t^{\text{state}}) \odot (1 + \gamma_t),$$

and passed to a transformer decoder with 4 layers and multi-head cross-attention over $\mathbf{Z}_t$. The decoder outputs latent action features

$$\mathbf{h}_k \in \mathbb{R}^{512},$$

which are mapped to joint-space actions via an MLP head

$$\pi : \mathbb{R}^{512} \to \mathbb{R}^{d_a}.$$

*2) Details on Simulation Augmented Representation Learning:* We describe the simulation-augmented representation learning framework, focusing on the auxiliary object decoder and its supervision.

**Shared visual encoder.** Real observations and simulation-augmented observations are processed by a shared visual encoder with identical weights. Given an observation at time $t$, the encoder outputs camera-level visual tokens

$$\mathbf{Z}_t \in \mathbb{R}^{N_{\text{cam}} \times d}, \quad d = 512.$$

The shared encoder allows supervisory signals from simulation to shape the visual representation used for real-world action prediction.

**Simulation object decoder.** For simulation-augmented samples, the encoded visual tokens are forwarded to an auxiliary object decoder. This decoder is used only during training and is not involved in action inference.

The camera tokens are flattened and concatenated,

$$\mathbf{z}_t^{\text{sim}} = \text{Flatten}(\mathbf{Z}_t) \in \mathbb{R}^{N_{\text{cam}} \cdot d},$$

and passed through a lightweight multilayer perceptron to predict the object pose

$$\hat{\mathbf{o}}_t = \left(\hat{\mathbf{p}}_t,\ \hat{\mathbf{M}}_t^{(6)}\right),$$

where $\hat{\mathbf{p}}_t \in \mathbb{R}^3$ is the object position and $\hat{\mathbf{M}}_t^{(6)} \in \mathbb{R}^6$ represents orientation.

For rotationally symmetric objects such as coins, orientation is represented using the normal-matrix

$$\mathbf{M}_t = \mathbf{n}_t \mathbf{n}_t^\top,$$

where $\mathbf{n}_t$ is the unit surface normal. We parameterize $\mathbf{M}_t$ using its six unique upper-triangular elements

$$\mathbf{M}_t^{(6)} = [M_{00}, M_{01}, M_{02}, M_{11}, M_{12}, M_{22}],$$

which is invariant to $\mathbf{n}_t \leftrightarrow -\mathbf{n}_t$ and avoids in-plane rotational ambiguity.

**Object pose loss (simulation data).** For simulation-augmented samples, the object decoder predicts object position $\hat{\mathbf{p}}_t \in \mathbb{R}^3$ and orientation $\hat{\mathbf{M}}_t^{(6)} \in \mathbb{R}^6$. We apply an $\ell_1$ loss on both components:

$$\mathcal{L}_{\text{obj}} = \|\hat{\mathbf{p}}_t - \mathbf{p}_t\|_1 + \left\|\hat{\mathbf{M}}_t^{(6)} - \mathbf{M}_t^{(6)}\right\|_1.$$

**Action loss (real data).** For real demonstrations, the action decoder predicts a sequence of joint-space actions $\hat{\mathbf{A}}_t \in \mathbb{R}^{T_p \times d_a}$. We supervise it using an $\ell_1$ regression loss:

$$\mathcal{L}_{\text{act}} = \left\|\hat{\mathbf{A}}_t - \mathbf{A}_t\right\|_1,$$

where $\mathbf{A}_t$ denotes the ground-truth action sequence from real demonstrations.

**Training pipeline.** Real demonstrations supervise action prediction, while simulation-augmented demonstrations supervise only object pose prediction. Both branches share the same visual encoder, and no simulated action labels are used. The full training procedure is shown in Alg. 1.

**Algorithm 1** Simulation-Augmented Representation Learning

```
Require: Real dataset $\mathcal{D}_{\text{real}}$, simulation dataset $\mathcal{D}_{\text{sim}}$
Preprocess $\mathcal{D}_{\text{sim}}$ to obtain augmented observations and object pose labels
while training do
    Sample real batch $\mathcal{B}_{\text{real}} \sim \mathcal{D}_{\text{real}}$
    Sample simulation batch $\mathcal{B}_{\text{sim}} \sim \mathcal{D}_{\text{sim}}$
    $\mathbf{Z}_{\text{real}} \leftarrow \text{Encoder}(\mathcal{B}_{\text{real}}.\text{obs})$
    $\hat{\mathbf{A}} \leftarrow \text{ActionDecoder}(\mathbf{Z}_{\text{real}})$
    $\mathcal{L}_{\text{act}} \leftarrow \|\hat{\mathbf{A}} - \mathcal{B}_{\text{real}}.\mathbf{A}\|_1$
    $\mathbf{Z}_{\text{sim}} \leftarrow \text{Encoder}(\mathcal{B}_{\text{sim}}.\text{obs})$
    $(\hat{\mathbf{p}}, \hat{\mathbf{M}}^{(6)}) \leftarrow \text{ObjectDecoder}(\mathbf{Z}_{\text{sim}})$
    $\mathcal{L}_{\text{obj}} \leftarrow \|\hat{\mathbf{p}} - \mathcal{B}_{\text{sim}}.\mathbf{p}\|_1 + \|\hat{\mathbf{M}}^{(6)} - \mathcal{B}_{\text{sim}}.\mathbf{M}^{(6)}\|_1$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{act}} + \lambda\, \mathcal{L}_{\text{obj}}$
    Update parameters using $\nabla \mathcal{L}$
end while
```

**Summary.** By supervising simulation data only through object-level position and symmetry-aware orientation signals, this framework avoids sim-to-real bias from erroneous action labels. At the same time, it encourages the shared visual encoder to capture geometry- and contact-relevant features that improve robustness to appearance variation in real-world manipulation.

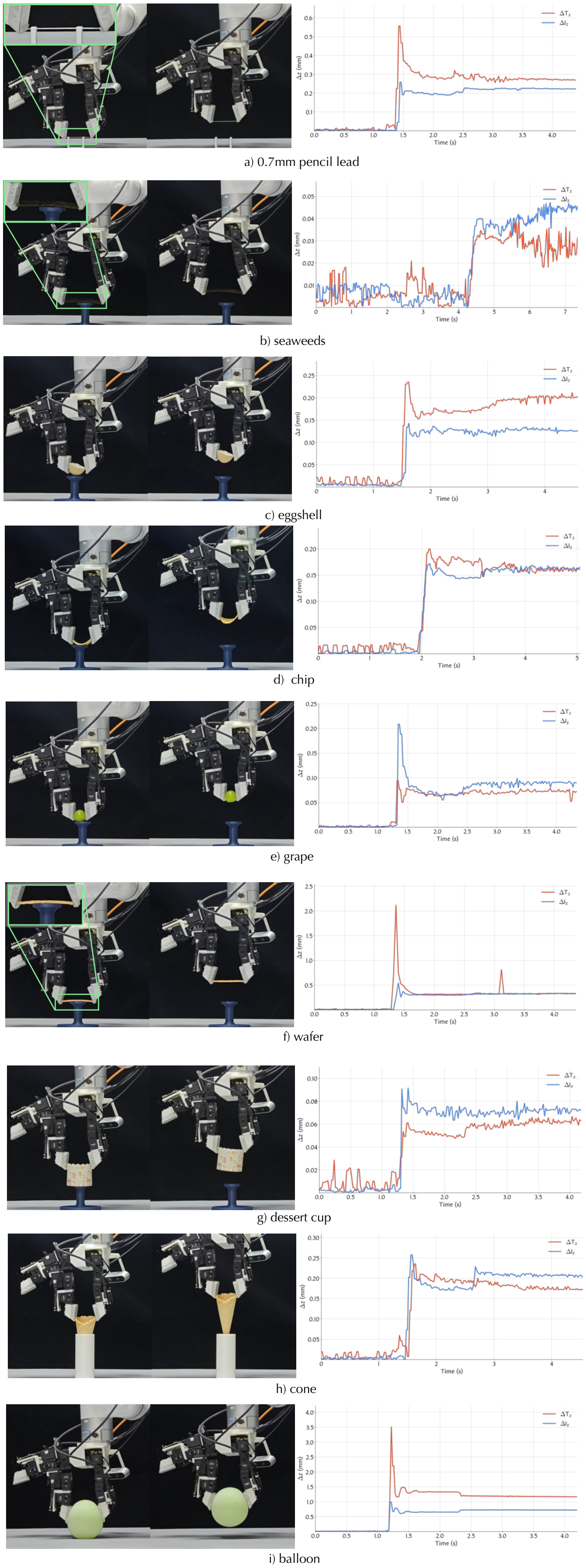
TABLE III: Domain randomization parameters

| Parameter | Prob. $p$ | Distribution |
|---|---|---|
| **Camera Image** | | |
| Brightness | 1.0 | $\sim U(-25,\ 25)$ |
| Contrast | 1.0 | $\sim U(0.5,\ 1.5)$ |
| Gaussian noise | 1.0 | $\sim \mathcal{N}(0,\ 40^2)$ |
| Crop | 1.0 | $\sim U(0.95,\ 1.05)$ |
| Horizontal translation | 1.0 | $\sim U(-0.05,\ 0.05)$ |
| Vertical translation | 1.0 | $\sim U(-0.05,\ 0.05)$ |
| Hue | 0.5 | $\sim U(-0.03,\ 0.03)$ |
| Saturation | 0.25 | $\sim U(0.5,\ 2.0)$ |
| Mean blur (root & wrist cam) | 1.0 | $7 \times 7$ average kernel |
| **Lighting** | | |
| Light intensity | 1.0 | $\sim U(2 \times 10^5,\ 5 \times 10^6)$ |
| Light color (per channel) | 1.0 | $\sim U(0.95,\ 1.05)$ |
| Light yaw rotation | 1.0 | $\sim U(-\pi,\ \pi)$ |
| **Material** | | |
| Floor gray | 1.0 | $\sim U(0.4, 1.0)$ |
| Skybox texture | 1.0 | $\sim U(\text{skybox set})$ |
| Coin color (per channel) | 1.0 | $\sim U(0, 1)$ |

*3) Details on Simulation Augmentation:* Our sim augmentation pipeline can be divided into three main categories: image, lighting, and material. Given that real-world cameras often exhibit various imperfections, we applied multiple post-processing augmentation methods—such as mean blur and Gaussian noise—to emulate these effects. In addition, we randomized lighting conditions, including light intensity and color, to cover as many real-world scenarios as possible. We also varied the color of the coins so that the policy learns to maintain consistent behavior under appearance changes. To further match real-world variations, we randomized the grayscale tone of the table surface to simulate different perceived colors under varying lighting conditions. The specific randomization parameters are listed in Table III.

### D. Additional Policy Learning Experiment Details

*1) Rollouts with Full ConSens Visualization:* The entire process of the policy rollout for our four experiments can be visualized in Fig. 21.



a) 0.7mm pencil lead

b) seaweeds

c) eggshell

d) chip

e) grape

f) wafer

g) dessert cup

h) cone

i) balloon

Fig. 20: Full Visualization of Force Curves

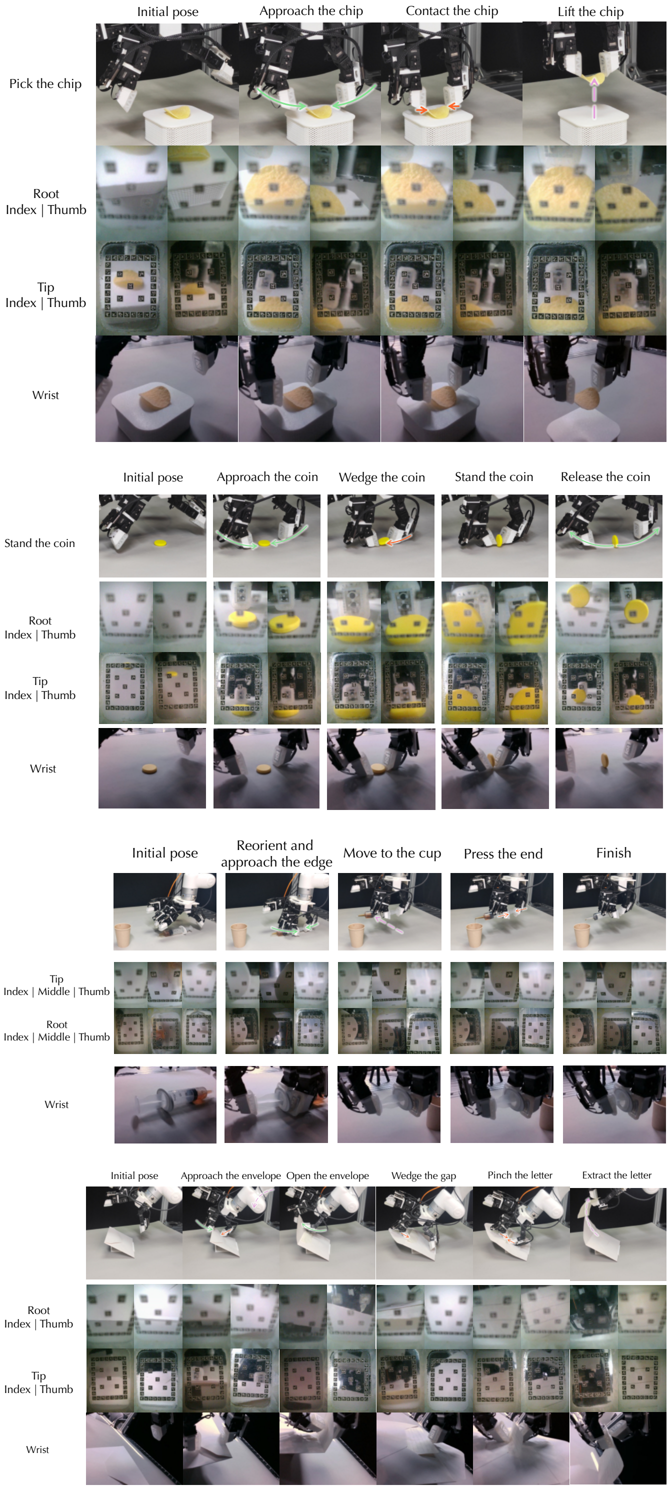


Fig. 21: **Visual task overviews.** Representative visual sequences for the four manipulation tasks evaluated in this work: chip picking, coin standing, syringe manipulation, and letter retrieving. Each row illustrates key interaction phases under both training and testing configurations.