ContactBench 1.0: A Benchmark for Everyday Contact-Rich Robotic Manipulation

Jingyi Xiang, Yixuan Wang, Jeannette Bohg, and Wenzhen Yuan

https://jingyi-xiang.github.io/contactbench



Fig. 1: We propose ContactBench 1.0, a benchmark for evaluating robot systems and policies on contact-rich manipulation. ContactBench 1.0 contains 36 tasks, more than 40 3D-printable assets, and their corresponding simulation components.

***Abstract*—Contact-rich manipulation underlies many everyday human activities, yet evaluating robots on such tasks remains difficult: the tasks involve diverse contact behaviors and are hard to reproduce across laboratories. Drawing on observations of everyday human manipulation, we identify eight essential skills that provide a practical framework for studying contact-rich manipulation. We introduce ContactBench 1.0, an extensible benchmark that instantiates these skills through 36 reference tasks and more than 40 object assets. The assets capture task-relevant geometry and contact behavior, can be fabricated reproducibly on consumer-grade 3D printers, and are each paired with a matching simulation model. Controlled task variations in both the real world and simulation enable evaluation of how well robot systems generalize. We evaluate recent representative vision-only and visual-tactile policy learning methods on six benchmark tasks. The results reveal that while vision-only policies struggle to perceive or react to contact events, visual-tactile policies do not meaningfully improve their performance, highlighting that contact-rich manipulation remains a significant challenge for current policy learning methods. By providing common task definitions alongside physical and simulated assets, ContactBench 1.0 offers a shared testbed for developing, evaluating, and comparing contact-rich manipulation systems. The benchmark will be open-sourced at https://jingyi-xiang.github.io/contactbench.**

***Index Terms*—Benchmark; Contact-rich Manipulation**

I. Introduction

Achieving human-level generality in robotic manipulation remains a long-standing goal of the robotics community. Recent progress in robot learning has produced increasingly capable systems, supported by large-scale demonstrations, expressive policy architectures, and broad task suites. Much of this progress has been demonstrated on tasks in which visually observable properties, such as object identity, appearance, and pose, provide strong guidance for action. Everyday manipulation, however, also requires robots to regulate forces, exploit environmental constraints, and respond to changes in contact. In these tasks, contact is the mechanism through which the robot accomplishes its goal. Developing robots that can effectively exploit contact is therefore essential to general-purpose manipulation.

This raises a core question: "What constitutes success in contact-rich manipulation?" Although contact-rich manipulation has been widely studied, many existing works focus on insertion and peg-in-hole tasks [1, 2, 3, 4, 5, 6] or tasks that require sustained force against a surface, such as wiping and peeling [7, 8, 9]. Everyday human activities, however, involve a much broader range of contact modes, such as grasping challenging objects with various environmental constraints, leveraging contact between the object held in hand and the environment, and manipulating articulated objects based on force feedback. Progress toward general-purpose manipulation therefore requires systematic, reproducible evaluation across a diverse set of tasks that cover this breadth of contact interaction.

We sought answers to this question by observing everyday human manipulation. These observations suggest that, despite differences in objects' appearances and functions, most manipulation sequences involving contact can be broadly described

Jingyi Xiang, Yixuan Wang, and Wenzhen Yuan are with the University of Illinois Urbana-Champaign, Urbana, IL, USA (e-mail: {jingyix4, yixuan41, yuanwz}@illinois.edu).

Jeannette Bohg is with Stanford University, Palo Alto, CA, USA (e-mail: bohg@stanford.edu).

in terms of eight skills: grasping thin or small objects, grasping multiple objects, grasping from clutter, placing under environmental constraints, in-hand repositioning, extrinsic contact control, articulated object manipulation, and insertion and alignment. Many complex manipulation tasks can be understood as combinations of these skills. We therefore believe that mastering these skills is essential for robots to perform a broad range of contact-rich manipulation tasks.

To support the community's efforts to develop these capabilities, we introduce ContactBench 1.0, an extensible benchmark that instantiates these skill categories through 36 reference tasks and more than 40 standardized object assets. The assets are designed to capture the functional contact interactions of everyday objects through their geometry and mechanical response. Each asset can be fabricated using 3D printing and is designed for reproducible fabrication on consumer-grade printers while preserving task-relevant contact behavior. Controlled variations in object geometry and internal mechanics support evaluation of policy generalization. Corresponding simulation models provide tunable resistance loads for studying related contact interactions in simulation. By providing shared task definitions and standardized physical and simulated assets, ContactBench 1.0 establishes common experimental ground for comparing methods across laboratories and supporting cumulative progress.

We demonstrate the utility of ContactBench 1.0 by evaluating recent vision-only policy learning methods (Diffusion Policy [10], $\pi_{0.5}$ [11]) and their tactile-augmented versions (Reactive Diffusion Policy [7], Foundation Tactile Policy [12]) on six tasks from our benchmark. The results reveal mixed performance across the evaluated methods. We observe that vision-only policies struggle to perceive and react to contact-dependent information that is difficult to infer from the vision modality alone. However, adding tactile input did not meaningfully improve performance for the policies we evaluated. Our analysis suggests that limited sensor coverage and difficulty translating tactile observations into appropriate actions may be the bottlenecks. These findings highlight challenges that extend beyond access to tactile sensing alone.

We open-source ContactBench 1.0 at https://jingyixiang.github.io/contactbench. The initial release focuses on rigid and articulated objects; deformable objects and objects with more challenging surface properties remain beyond its current scope, but will be a potential future extension direction. We envision ContactBench 1.0 as a growing community resource that can incorporate additional tasks, assets, and skills over time. By providing shared infrastructure and a systematic account of contact-rich skills, we aim to make physical interaction a measurable dimension of general-purpose robotic manipulation.

II. Related Work

A. Reproducible Physical Assets and Benchmarks

Physical benchmarks pursue reproducibility in two main ways. Some standardize the hardware: YCB [13] provides everyday objects with geometric models, NIST task boards [14] define small-parts assembly tasks and protocols, and ManipulationNet [15] supports distributed real-world evaluation with standardized hardware kits. Others release printable designs that any lab can fabricate: FurnitureBench [16] pairs printable furniture with demonstrations and a simulator for long-horizon assembly, Fabrica [17] benchmarks dual-arm multi-part assembly, FMB [18] evaluates functional manipulation skills such as grasping, repositioning, and insertion, while POMDAR [19] introduces a benchmark for dexterity. Recent work also varies task conditions in a controlled way. MatchMaker [20] generates mating parts with user-specified clearances, and Peg-in-Bench [6] varies peg geometry, insertion tolerance, spatial layout, and assembly structure. BusyBox [21] instead varies appearance, rearranging modules of buttons, switches, sliders, and other controls while preserving their affordances. Many of these tasks involve assembly or insertion, and the forces they require are either fixed by the asset or set by part clearance. Our benchmark spans a broader range of everyday contact-rich skills, and its printed compliant springs set the required force independently of fit and appearance.

B. Simulation Benchmarks for Manipulation

Simulation benchmarks support reproducible comparisons across tasks and learning methods. RLBench [22], Meta-World [23], and robosuite [24] provide standardized task suites, ManiSkill2 [25] adds wide object variation, and ManiSkill3 [26] parallelizes simulation and rendering on the GPU. CALVIN [27] and LIBERO [28] extend evaluation to long-horizon language-conditioned tasks and lifelong learning. Other environments focus on particular interactions. UniDoorManip [29] simulates doors with diverse geometries and mechanisms. Several simulation platforms additionally support tactile sensing: Taxim [30] renders vision-based tactile images, Tactile Gym 2.0 [31] compares optical tactile sensors through sim-to-real reinforcement learning, Robot Tactile Olympiad [32] benchmarks tactile reinforcement learning on four dexterous hands, while Manifeel [33], UniVTac [34], and OpenViTac [35] offer standardized benchmarking platforms suitable for evaluating a wide range of visual-tactile policies. None of these environments, however, pairs its tasks with physical assets whose contact behaviors can be varied. Our simulation suite mirrors each printable mechanism in the benchmark and exposes that contact behavior as an explicit tunable parameter.

III. ContactBench 1.0

Drawing on observations of everyday human manipulation, we identify eight key skills that underlie a broad range of contact-rich manipulation tasks. To enable reproducible evaluation of these skills across laboratories, we introduce ContactBench 1.0. The benchmark comprises 36 reference tasks organized around the eight skills and more than 40 standardized, 3D-printable assets with corresponding simulation models. Shared task definitions, explicit success criteria, and reproducible assets provide a common basis for comparing robot systems. To evaluate generalization, we introduce controlled task and asset variations spanning both visually observable changes, such as object geometry, and changes that are not readily apparent from vision alone, such as contact



Fig. 2: The eight contact-rich manipulation skills we identified for everyday contact-rich manipulation tasks. We built ContactBench 1.0 to instantiate these skills.

forces. These variations enable evaluation of how manipulation systems respond to changes that require adaptations in behaviors. The simulation counterparts of the assets and tasks will be introduced in Section IV.

A. *Contact-Rich Manipulation Skills*

The eight skills we identified (shown in Figure 2) are drawn from observations of routine human activities and are chosen to span a broad range of contact-rich behaviors rather than to form a comprehensive or mutually exclusive taxonomy. Typically, humans use a combination of those skills to complete multi-stage tasks. We describe each skill in detail below.

Grasping thin or small objects. This skill involves acquiring a grasp on objects whose thin or small form factor demands more precision than objects of typical size and shape. Although pick-and-place has been studied extensively, the objects involved are often easy to grasp and require few considerations about grasp strategy. In contrast, many everyday objects, such as cards and keys, require not only precise grasp acquisition but also a carefully chosen grasp strategy, so that the resulting grasp supports the intended downstream manipulation. For example, a top-down grasp on a card's thin edges is unstable and makes subsequent swiping or insertion actions difficult, whereas a grasp on its two faces is stable and well suited to these tasks. Achieving the latter often requires in-hand repositioning or a dedicated grasping strategy, such as sliding the card to the table edge to expose both faces, or using the thumb as a pivot while the index finger lifts the card off the surface.

Grasping multiple objects. This skill involves acquiring and holding several objects in a single grasp so that they can be transported together. It can substantially improve the efficiency of many everyday tasks: when clearing loose screws from a workspace, for example, collecting several in hand and depositing them into a bin together is much faster than picking and placing each one individually. Grasping multiple objects requires the end-effector to form and maintain an enclosure that keeps previously picked objects secure while acquiring new ones, which is a requirement that is difficult for parallel-jaw grippers and often benefits from high-dexterity end-effectors. Despite its practical value, multi-object grasping remains relatively unexplored in many policy learning approaches, where methods often assume a single object per grasp [36, 37, 38].

Grasping from clutter. This skill involves acquiring a grasp on a target object that is surrounded by and in contact with other objects. Objects in everyday environments are rarely presented in isolation: utensils sit mixed together in a drawer, screws and nuts of the same size share a storage bin, keys hang together on a key ring, cups tightly stack on top of each other to save storage space. Retrieving a target object in these settings requires locating it and then separating it from its neighbors, a process we refer to as singulation, without disturbing the scene more than necessary. The suitable manipulation strategy depends on how the clutter is arranged. In a loose clutter, such as a bin of screws, singulation often involves burrowing the gripper fingers into the pile to form a secure grasp on the target. In a tightly organized clutter, such as a deck of cards or a stack of cups, the target object must instead be extracted without displacing its neighbors.

Placing under environmental constraints. This skill involves placing an object where the surrounding environment constrains how it must be positioned and released. Although pick-and-place has been studied extensively, placement is usually treated as dropping an object anywhere in a broad target region. Many everyday placements are more demanding: hanging requires precise actions to align the object with the hanging hook, stacking requires releasing the object without disturbing the stack beneath it, and placing into a cluttered region requires finding an unoccupied gap and depositing the object into it. In each case, the final alignment and release must be carefully controlled, since a premature or misaligned release fails the task even if the preceding approach was correct. We therefore treat constrained placement as a skill in its own right, rather than as the trivial final step of pick-and-place.

In-hand repositioning. This skill involves changing how a held object is grasped, or its pose relative to the end-effector, without setting the object down. It is a routine part of human manipulation that bridges object acquisition and downstream usage, since an object is rarely picked up in the grasp configuration that the downstream task requires. Sometimes the grasp type itself needs to change [39]: an Allen key might be most easily picked up with a pinch grasp, but using it to drive a screw usually requires a different grasp. Other times only the object pose needs to be adjusted: a key or USB stick picked up in the wrong orientation must be flipped before insertion. In-hand repositioning is most versatile with multi-fingered hands; parallel-jaw grippers are largely limited to sliding the object within the grasp, often relying on extrinsic contact with the environment [40, 41, 42]. While many prior works on dexterous in-hand manipulation focus on continuous rotation of blocky or spherical objects [43, 44, 45, 46, 47], the objects humans most often reposition, such as keys, cards, tools, and USB sticks, have more challenging form factors and

remain difficult to effectively manipulate in-hand [48].

Articulated object manipulation. This skill involves operating objects whose rigid parts are connected by joints that constrain their relative motion. Such objects are abundant in human environments, including doors, drawers, buttons, switches, faucets, and levers. Therefore, operating articulated objects is a basic requirement for any robot working in human spaces. The defining challenge is complying with the kinematic constraint. A rigid position-controlled trajectory that wrongly estimates the kinematic constraint can rapidly build up damaging force on both the manipulator and the object. Therefore, effective manipulation of articulated objects must either estimate the constraint accurately or yield to it compliantly.

Extrinsic contact control. This skill involves regulating extrinsic contact, the contact made between an in-hand object and the environment, as opposed to intrinsic contact made between the gripper and the object it holds. It is especially prevalent in tool use: wiping a whiteboard with an eraser requires regulated pressure between the eraser and the board, and driving a screw with an Allen key requires maintaining the contact between the Allen key tip and the screw socket while transmitting torque. The defining challenge is that extrinsic contact is not a rigid joint. Without active regulation, it can slip, disengage, or transmit the wrong force. This distinguishes extrinsic contact control from articulated object manipulation, where the object's joints maintain the constraint mechanically.

Insertion and alignment. This skill involves bringing two parts into precise alignment and full engagement. Insertion has been widely studied as a canonical contact-rich problem with tasks such as peg-in-hole, electronics plugging, and gear meshing. It captures a distinct combination of challenges that involve tight tolerances, constrained insertion direction, and often substantial insertion force. Following these characteristics, we extended the category to include the initial alignment phase of threaded assembly, which involves similar tight tolerances for engaging the threads. Insertion can be further complicated by occlusion caused by the manipulated objects themselves: the pins of a charging block can be occluded by the charging block body, and the tip of a screw can be occluded by the nut being threaded onto it. The robot must therefore reason about geometry and contact it cannot directly observe visually.

B. Benchmark Tasks

To evaluate the contact-rich skills described above, we provide 36 reference tasks with explicit task definitions and success criteria. As shown in Figure 3, the task set comprises 31 short-horizon tasks, each requiring one or two skills, and five long-horizon tasks that combine multiple skills across several stages. Since many of these tasks remain challenging for current robot systems, we record humans completing them to show what successful execution looks like. We intend these tasks to be attempted by any suitable robot embodiment. Together, these tasks expose open challenges for current robot systems and provide a common benchmark for measuring progress toward solving them. Table II presents the six short-horizon tasks we used in our benchmarking experiments alongside their detailed specs, and the complete task list is shown in Appendix C.

TABLE I: Task variations supported by our benchmark. Blue rows denote behavioral variations that affect task behavior but are ambiguous to vision; green rows denote visual + behavioral variations that affect both task behavior and appearance. Terminology follows Star-Gen [49].

Task	Variations
Turn doorknob to open door Turn door handle to open door	Required rotation amount Rotation force Object geometry
Plug in charger Plug in USB	Insertion force Object geometry
Put bottle cap onto bottle Install lightbulb	Required rotation amount Object geometry
Tighten screw with Allen key Tighten screw with screwdriver Tighten nut with wrench Turn key to unlock door	Required rotation amount
Grasp target key from keychain Retrieve single utensil from bin Retrieve single fastener from bin Collect utensils Collect fasteners Place utensil into bin	Object geometry Number of objects
Pick up object from platform Hang object In-hand repose In-hand grasp transition Turn dial to target position	Object geometry
Retrieve top cup from stack Retrieve single card from deck Place card into holder Stack cups	Number of objects

C. Benchmark Task Variations

We introduce task variations designed to evaluate behavioral generalization [49], which is the ability of a policy to adapt to novel task-relevant conditions that are ambiguous to vision. While recent work on generalist manipulation policies has demonstrated impressive zero-shot transfer to novel scenes, objects, and language instructions, behavioral generalization has not received as much attention, despite being especially prevalent in contact-rich manipulation. Humans can unlock a door regardless of whether the doorknob must be turned $45°$ or $60°$, plug in a charger regardless of whether the outlet is tight or loose, and tighten a bottle cap regardless of whether it takes two or three revolutions. Each of these variations is difficult to perceive visually but is central to successful task execution. Following the generalization taxonomy proposed in Star-Gen [49], the perturbations we consider fall into two categories: behavioral perturbations, which affect



Fig. 3: ContactBench 1.0 includes 36 manipulation tasks to concretely evaluate a robot system's proficiency on the eight skills. Many tasks require more than one skill to complete. We record humans solving the tasks to illustrate successful task executions due to the difficulty of many tasks for current robot systems.

the behavior required to complete a task without altering its visual appearance, and visual + behavioral perturbations, which affect both the task behavior and visual appearance. Several standard perturbations, such as changes to object color, lighting, camera viewpoint, and initial object pose, are supported by our benchmark without dedicated asset designs, and we do not enumerate them here.

Table I lists supported benchmark task variations, including but not limited to required interaction forces, required rotation amounts, and object geometry. Figure 5(c) shows examples of the various levels of interaction forces available as task variants, which are achieved by deliberate asset designs described in Section III-D. Many reference tasks support at least one variation, and several support both behavioral and visual + behavioral variations, allowing the same task to be evaluated under either or both axes.

D. Benchmark Assets

ContactBench contains more than 40 assets that support the evaluation of the manipulation skills and generalization capabilities discussed in Section III-A and Section III-C. Compared to benchmarks that standardize the assets through specific vendors and distributors [13, 14, 15], we prioritize the accessibility of our assets to researchers worldwide. All assets are therefore 3D-printable in standard polylactic acid (PLA) filament, which is inexpensive and easy to print, and are assembled with standard button-head M3 screws and nuts.

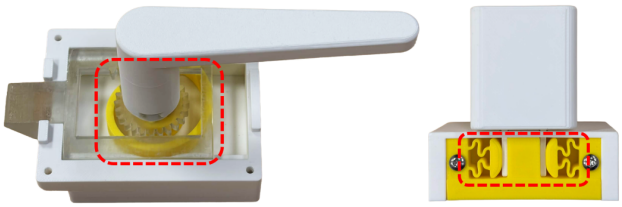


Fig. 4: We design spring structures inside the assets to approximate their resistance force during manipulation. Here is an example of the torsional spring inside a door handle and a compression spring inside an outlet.

Asset files, print settings, and assembly instructions will be made available on the project website.

The accessibility of 3D printing comes with two challenges: how realistically the printed parts reproduce the contact forces of real objects, and how consistently they do so across different 3D printers. Existing 3D-printable benchmarks for contact-rich manipulation often use large clearances between mating parts, resulting in loose fits that generate little contact force [16, 18, 17]. This contradicts the demands of contact-rich manipulation, where applying and regulating contact forces is central to task execution. Tightening the clearance between parts can increase frictional contact forces, but printing accuracy varies across printers, so the forces obtained with a given clearance on one printer might not reliably transfer to another.

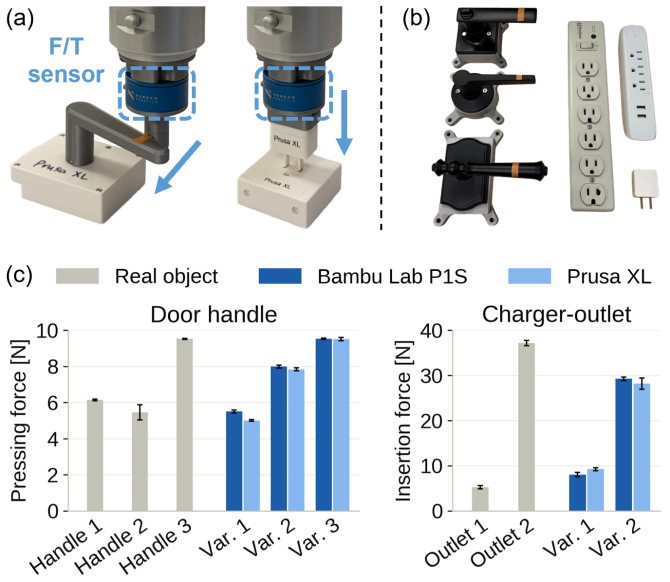


Fig. 5: The resistance forces exhibited by our assets are comparable to those of real objects and are generally consistent across different 3D printers. We show experimental results over a door handle and a charger-outlet pair with their variants. (a) shows the experimental setup, (b) shows the real products used for comparisons, and (c) shows the measured max pressing and insertion forces across the real objects and the asset variants printed by two printers.

We address both challenges by reducing the dependency of force generation on printing accuracy. Rather than relying on a tuned fit between parts, our assets exploit the PLA material compliance and generate interaction forces with compliant spring designs, such as the torsion spring in the door handle and the compression spring in the outlet shown in Figure 4. Since a spring's behavior is governed by its geometry and the material properties of PLA, the resulting forces depend on quantities that are more consistent across printers than part tolerances. Varying the spring geometry also lets us provide task variants that require different forces to complete, as described in Section III-C.

To validate our approach, we mount a force/torque (F/T) sensor on the robot end-effector and let the robot execute preprogrammed motions that press a door handle and insert a charger into an outlet, as illustrated in Figure 5(a). As references, we first measure the forces produced by the three real door handles and two real outlets shown in Figure 5(b). We then print three variants of our door handle and two variants of our charger-outlet assets on a Bambu Lab P1S and a Prusa XL and repeat the measurements. Both printers use the same filament and print settings. The force measurement is repeated ten times on each real and printed asset. As shown in Figure 5(c), the printed door handles produce 5.01N – 9.54N pressing forces, matching closely with the 5.45N – 9.53N produced by the real handles. The printed outlets produce 8.04N – 29.26N insertion forces, which fall within the 5.24N – 37.18 N range spanned by the real outlets. Across all tested variants, the forces generated by assets printed on the two printers differ by at most 0.5N for the door handle and 1.19N for the charger-outlet pair. These results indicate that our asset designs with compliant PLA springs produce interaction forces comparable to real objects and can achieve this reproducibly across printers.

IV. Simulation Benchmark

We develop the corresponding assets and task setups for ContactBench 1.0 in simulation, building on the Newton physics engine [50] with the MuJoCo Warp solver [51, 52]. The simulated assets reproduce the behavior of the real assets with an emphasis on force-realistic contact interactions: spring resistance, frictional retention, constrained motion, geometric interlocking, and changes in load during engagement and release. The simulation suite supports demonstration collection through scripted policies and teleoperation, and includes an optional module for simulating a fingertip GelSight tactile sensor. Figure 6 provides an overview of the simulation suite.

A. Assets and contact modeling

We export the source CAD of the assets as separate rigid parts and assemble mechanisms using their joint axes, travel limits, and fixed supports. Printed spring bodies are replaced by prismatic or revolute spring–damper drives, exposing stiffness and preload as explicit parameters. This retains the mechanism's functional resistance without simulating the printed spring's full deformation. Figure 7 shows the force response of one asset, the door handle, with its printed spring modeled as a revolute spring with stiffness K and preload τ_0. Tuning these two parameters reproduces the measured profiles of handles printed with three different springs.

To balance the contact fidelity and computational efficiency, we adopt different strategies to model each asset. Contact-critical geometry, including threads and insertion passages, uses mesh signed-distance fields (SDFs); simpler collision primitives are used where detailed geometry is unnecessary. Thin features require sufficient SDF resolution and surface sampling. For example, the card-grasping implementation uses a 512-cell maximum SDF resolution for its 1.2 mm-thick card. Sub-resolution features that require geometric approximations are documented per task. Appendix A-A shows how these modeling choices apply to representative mechanisms.

B. Tactile sensing simulation

To support potential tactile-augmented policy learning, we simulate GelSight Mini sensors mounted on the gripper fingers, including their compliant sensing pads and rigid housings. Contact is modeled using a hydroelastic pressure-field framework [53]. The contact surface is rasterized into spatial pressure maps, with pressure scaling based on the normal load resolved by the dynamics solver. Sensor indentation is estimated through a nonlinear pressure–indentation relation that captures increased resistance under compression. In the observation model, relative tangential motion is accumulated into a shear displacement field, bounded by local pressure and friction.

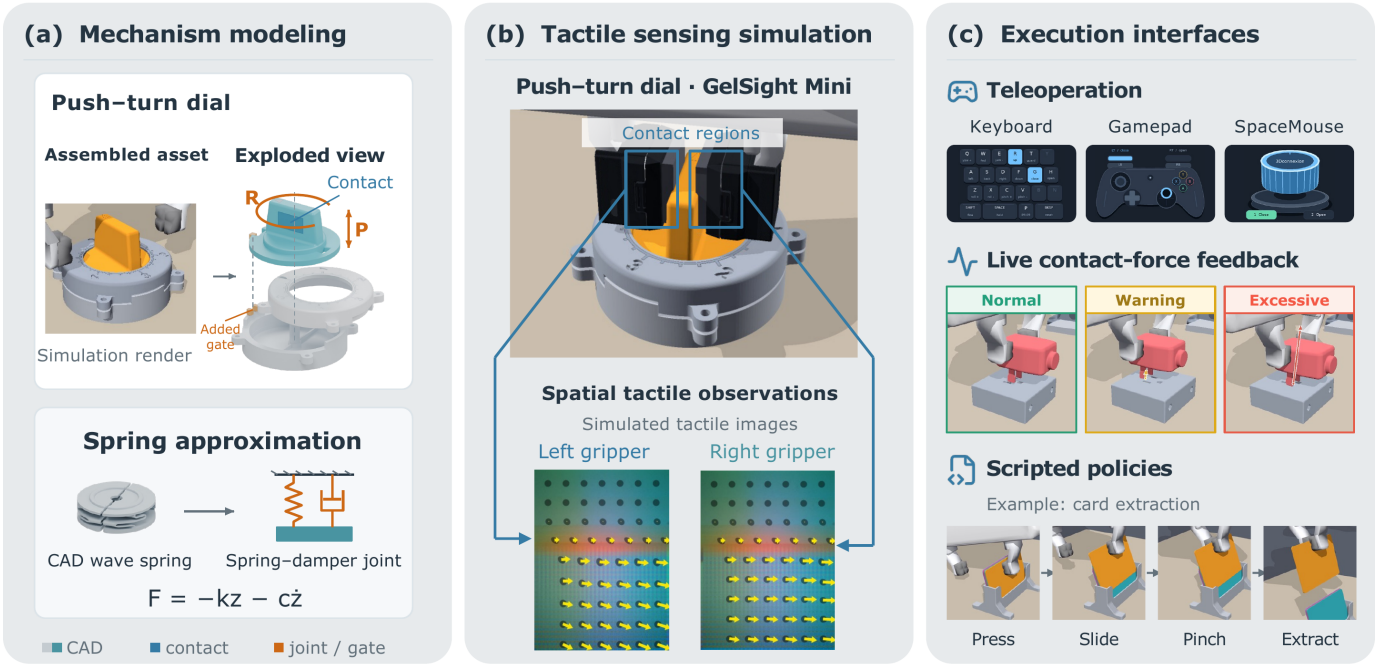


Fig. 6: Overview of the ContactBench 1.0 simulation suite. The suite provides assets that match the 3D-printable versions in terms of shape and mechanisms **(a)**, a simulated GelSight Mini interface **(b)**, and a teleoperation interface for demonstrating manipulation policies through keyboard, gamepad, or SpaceMouse **(c)**. We also provide a set of scripted policies for the contact-rich manipulation tasks.

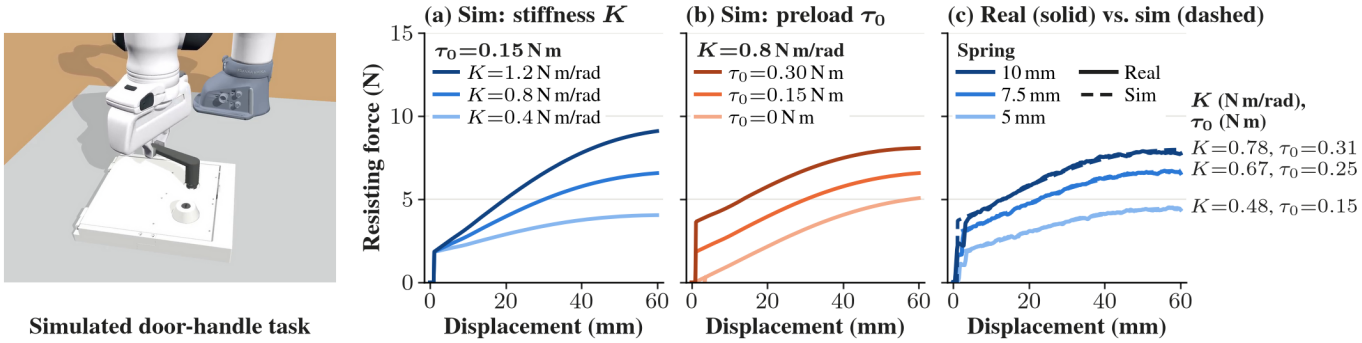


Fig. 7: Our simulated asset models feature tunable force response. In this example of the door handle asset, we make a robot rotate it at a linear speed of 10 mm/s. Here we show the measured force/torque value at the robot wrist. **(a)** Force response when changing the simulated door handle's parameter of revolute spring with stiffness K. **(b)** Force response when changing the simulated door handle's parameter of pre-load torque τ_0. **(c)** Force response of simulated assets with varied settings in comparison with the response from real assets. The real ones are printed on a Bambu Lab P1S with 5, 7.5, and 10 mm springs (solid, mean of 10 trials). The simulation reproduces the shape and ordering of the measured profiles.

The resulting deformation fields are rendered into tactile images using optical shading based on Taxim [30]. Local surface gradients and image position determine color intensity, while marker displacement reflects shear deformation and changes in surface height. The module provides tactile images, pressure and indentation maps, shear displacement, and net finger forces. These outputs describe the spatial structure of contact and its evolution during manipulation.

C. Teleoperation and data collection

We provide teleoperation interfaces for keyboard, gamepad, and SpaceMouse to support interacting with the simulated assets and task setups. All three devices control end-effector translation, rotation, and gripper opening through a shared command interface, with adjustable speed and a fine-control mode for precise alignment. Live contact-force feedback allows operators to monitor loading and adjust their commands during execution (Figure 6(c)). Collected trajectories align executed actions with observations and contact measurements,

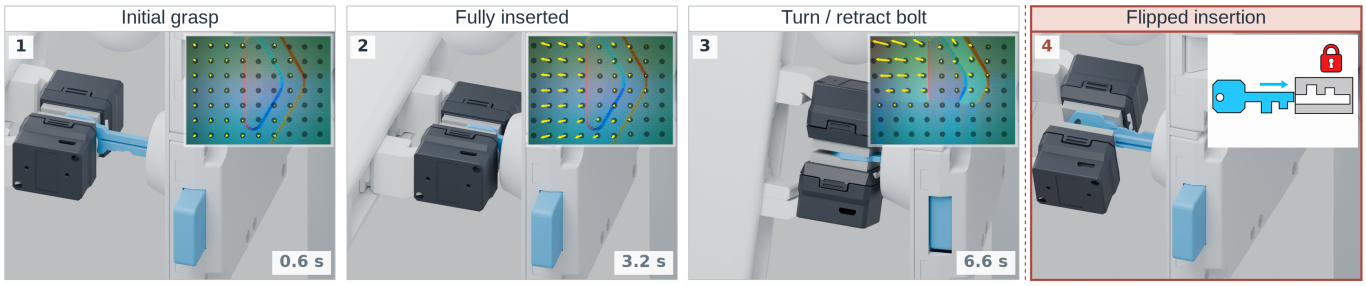


Fig. 8: Example of the scripted policy in simulation with tactile feedback: key insertion and turning. The correctly oriented key is inserted and turned to retract the bolt (1–3). The keyway interlock rejects a flipped-key insertion attempt (4). Insets show simulated tactile images from the right fingertip; yellow arrows indicate marker displacement.

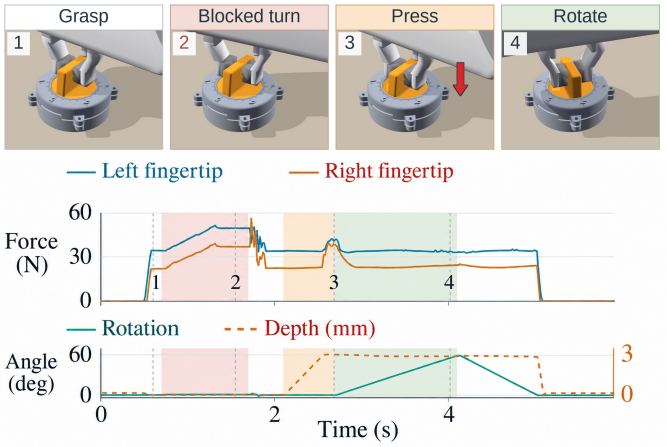


Fig. 9: Example of the scripted policy in simulation: push-and-turn dial operation. The plots show fingertip forces, dial rotation, and axial depression during the process, with numbered guides matching the keyframes. The red, orange, and green bands denote the phase of blocked turning, pressing, and permitted rotation, respectively.

while retaining task outcomes and simulation configurations for potential policy training and evaluation.

D. Scripted policies and task execution

We provide task-specific scripts to demonstrate the intended interaction sequences in simulation and generate reference trajectories. Each script specifies a predefined sequence of motion phases, such as approach, grasp, engagement, manipulation, and release. Depending on the task, execution follows a timed trajectory or updates motion targets using simulated object states and contact-force measurements. The successful rollouts can be recorded to generate seed demonstrations for potential subsequent policy learning.

Figure 8 and Figure 9 show two examples: key-operated unlocking with simulated fingertip tactile images, and a push-and-turn dial that rotates only after it is pressed down.

V. Experiments

ContactBench 1.0 is designed to evaluate different manipulation systems, where the performance provides insight into the system's deficiencies in addressing contact-rich manipulation. In this section, we demonstrate the use of the real-world benchmark sets with four recent policy learning methods on six tasks, and analyze their behaviors. We expect the results to provide insight for the following questions:

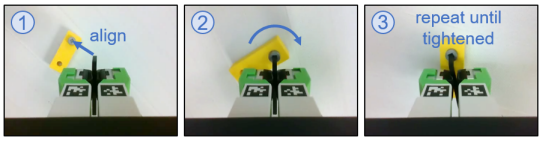
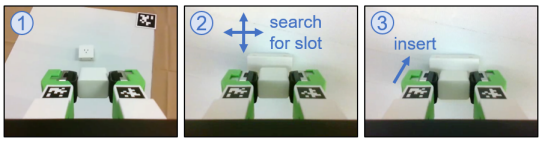
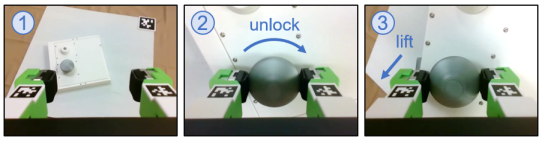
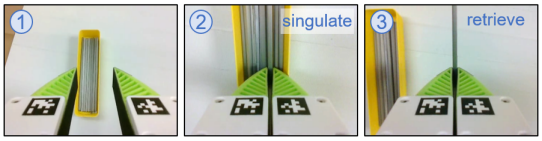
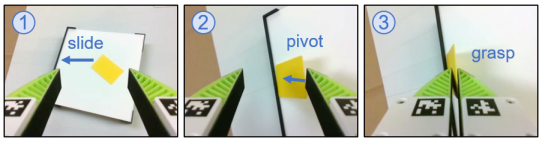
1) What are the major challenges for vision-only policies to perform contact-rich tasks?
2) Do representative visual-tactile policies address these challenges?
3) How do the evaluated policies generalize when task-relevant conditions change in ways that are difficult to perceive visually?

The policies we evaluate include two vision-only policy learning methods, Diffusion Policy (DP) [10] and $\pi_{0.5}$ [11], and their representative tactile-augmented variants, Reactive Diffusion Policy (RDP) [7] and Foundation Tactile Policy (FTP-1) [12]. The performance of each policy across the selected tasks is shown in Table III and Figure 11. While the policies were able to make progress toward solving each task, the overall success rate is 56.7% for the best policy-task combination, with the turn doorknob to open door task being the exception where all policies achieved near 100% success rate. We found that perceiving and reacting to contact events remains challenging for vision-only policies in the form of establishing and maintaining contact, precise alignment, avoiding unsafe forceful interactions, and recognizing task-critical states. These challenges contribute to policies' inability to complete the tasks, taking noticeably longer and requiring more attempts to complete certain tasks compared to the human demonstrator, and generating excessive forces between the end-effector and the environment. The two representative visual-tactile policy learning methods we evaluated do not systematically address these challenges faced by vision-only policies, and oftentimes performed worse. On the other hand, in the generalization experiments, we found that vision-only policies showed fragility under behavioral perturbations, while visual-tactile policies adapted better.

A. Experiment Setup

Policies. We benchmark two vision-only policy learning methods together with their tactile-augmented variants. Diffusion Policy (DP) [10] models robot action sequence generation as a conditional denoising diffusion process, and has become

TABLE II: Benchmark tasks used for policy learning experiments

Task Execution Sequence Using UMI	Task Specs
	**Task:** Thread nut onto screw **Skills:** Insertion and alignment, articulated object manipulation **Description:** Screw the nut held in hand onto the screw head placed in the workspace such that their threads fully engage. **Success criteria:** The nut and screw threads engage. The robot movement stops after terminating the rotation motion, moving away from, and breaking contact with task objects.
	**Task:** Tighten screw with Allen key **Skills:** Extrinsic contact control, insertion and alignment **Description:** A screw is placed into the screw slot of an assembly block in the workspace, but the threads are unengaged. Use the Allen key grasped in hand to tighten the screw. **Success criteria:** The screw is tightened and remains stationary when a small force is applied in an attempt to rotate the screw. The robot movement stops after terminating the tightening motion, moving away from, and breaking contact with the screw.
	**Task:** Plug in charger **Skills:** Insertion and alignment **Description:** An outlet is placed in the workspace. Fully insert the charger held in hand into the outlet. **Success criteria:** The charger is fully inserted into the outlet, such that it does not noticeably move when pushed further toward the outlet. The robot movement stops after terminating the insertion motion, moving away from, and breaking contact with task objects.
	**Task:** Turn doorknob to open door **Skills:** Articulated object manipulation **Description:** Turn the doorknob fully to unlock the door, then open the door. **Success criteria:** The door is opened by at least 5 degrees, such that it is visually obvious that the door has been unlocked.
	**Task:** Retrieve single utensil from bin **Skills:** Grasping from clutter, grasping thin and small objects **Description:** A utensil storage bin is placed in the workspace. The target utensil is placed in the bin alongside multiple other utensils. Pick up a single copy of the target utensil from the cluttered storage bin and place it on the table without removing other objects from the bin. **Success criteria:** The robot movement stops after separating a single copy of the target utensil and placing it on the table. Other utensils should remain in the storage bin. The robot should not remain in contact with the task objects.
	**Task:** Pick up card from platform **Skills:** Grasping thin and small objects **Description:** Pick up the thin and/or small object placed on the flat platform. **Success criteria:** The target object is stably grasped by the robot and lifted off the platform.

one of the most widely adopted policy learning frameworks in manipulation research, serving as the backbone and comparison point for a large body of subsequent work. $\pi_{0.5}$ [11] is a vision-language-action model that combines flow-matching action generation with vision-language pretraining, and is representative of the recent wave of large-scale generalist manipulation policies. Their tactile-augmented variants, Reactive Diffusion Policy (RDP) [7] and Foundation Tactile Policy (FTP-1) [12], extend the architecture to incorporate tactile feedback. Reactive Diffusion Policy uses a slow-fast architecture with a tactile-conditioned reactive policy on top of Diffusion Policy, and FTP-1 is a generalist tactile policy built on $\pi_{0.5}$ and pretrained on data from diverse tactile sensors and robot embodiments.

We collect training data with a customized Universal Manipulation Interface (UMI) [54] hand-held gripper. A single wrist camera view serves as the RGB camera input for all policies, while the marker displacements and tactile images from the GelSight Mini sensors are used as tactile inputs for RDP and FTP-1 respectively. We use the official implementations for all policies and train them to predict future actions relative to the current robot pose. We focus on evaluating a policy's ability to learn fine-grained, contact-rich manipulation skills. Therefore, we finetune $\pi_{0.5}$ and FTP-1 on each selected task and effectively treat them as single-task specialist policies, leaving evaluation of their reasoning and language under-

TABLE III: Policy performance on the six selected tasks (30 episodes per method per task)

Task	Human	DP				RDP				$\pi_{0.5}$				FTP-1			
	Time	SR ↑	Prog ↑	>40 N ↓	Time ↓	SR ↑	Prog ↑	>40 N ↓	Time ↓	SR ↑	Prog ↑	>40 N ↓	Time ↓	SR ↑	Prog ↑	>40 N ↓	Time ↓
Thread nut onto screw	17.5 s	8/30	0.39	33%	1.72×	0/30	0.24	27%	–	**14/30**	**0.66**	27%	1.22×	**14/30**	0.63	83%	1.32×
Tighten screw with Allen key	19.7 s	0/30	0.58	10%	–	0/30	0.36	0%	–	**17/30**	**0.82**	7%	1.50×	3/30	0.45	3%	0.84×†
Plug in charger	7.3 s	2/30	0.32	80%	1.06×†	4/30	0.34	83%	1.15×†	**6/30**	**0.44**	40%	0.89×	4/30	0.35	40%	0.95×†
Turn doorknob to open door	7.6 s	**30/30**	**1.00**	10%	0.92×	29/30	0.98	63%	0.91×	**30/30**	**1.00**	7%	0.87×	29/30	0.98	43%	0.89×
Retrieve single utensil from bin	11.8 s	0/30	0.50	97%	–	*not evaluated*				**16/30**	**0.70**	47%	2.05×	*not evaluated*			
Pick up thin card	7.5 s	**14/30**	**0.74**	57%	1.11×	*not evaluated*				2/30	0.28	60%	0.93×†	*not evaluated*			

SR: number of successful episodes out of 30. Prog: mean fraction of task checkpoints reached per episode (0–1). >40 N: percentage of episodes with peak wrist force exceeding 40 N. Time: mean completion time of successful episodes; for the policies, shown as a multiple of the time taken by human demonstrations. †Fewer than 5 successful episodes. A dash (–) marks no successful episodes.

standing capabilities in the contact-rich setting to future work. Policy training details and hyperparameters can be found in Appendix B.

Tasks. We select six tasks from the benchmark for policy evaluation: *thread nut onto screw*, *tighten screw with Allen key*, *plug in charger*, *turn doorknob to open door*, *retrieve single utensil from bin*, and *pick up card from platform*. Table II shows the detailed specs for each task and their typical UMI demonstration sequence. Together, these tasks cover five of the six skills introduced in Section III-A that a single robot arm with a parallel-jaw gripper can complete. We selected them because we believe they are among the most challenging tasks in the benchmark that are feasible for our robot setup.

Robot hardware setup. All experiments use a UR5e robot equipped with a WSG50 parallel-jaw gripper and a RealSense D435i wrist camera. For *thread nut onto screw*, *tighten screw with Allen key*, *plug in charger*, and *turn doorknob to open door*, we use custom 3D-printed fingers equipped with GelSight Mini tactile sensors. We cover the sensing area of each GelSight Mini with a layer of 3M gripper material to reduce gel pad wear and tear. For *retrieve single utensil from bin* and *pick up card from platform*, we use compliant Fin-Ray fingers instead, since these tasks frequently generate contact that is outside of GelSight Mini's fingertip-only sensing coverage and GelSight Mini's bulky form factor is not well suited to these tasks. Consequently, tactile signals are not recorded during demonstrations for these two tasks, and only the vision-only policies are evaluated on them.

Demonstration collection. All demonstrations were collected with a customized UMI hand-held gripper. To achieve the high trajectory tracking accuracy required for contact-rich manipulation, we replace the wide-angle fisheye camera with a RealSense D435i camera, and the SLAM-based localization with HTC VIVE-based motion tracking using OpenVR [55]. We calibrate the HTC VIVE tracking system to collect demonstration trajectories in the robot base frame. We prevent the human operator from relying on depth perception or third-person visual cues that the policies would not have access to during rollouts by requiring them to complete the tasks based on the wrist camera stream and their own tactile feedback. In principle, this matches the observations available to tactile-augmented policies at rollout time. We note, however, that human tactile perception remains richer than what the GelSight Mini tactile sensors can capture. While GelSight Mini provides high-resolution contact information only within the fingertip contact patch, the human hand holding the UMI gripper can perceive contact happening across the entire gripper body.

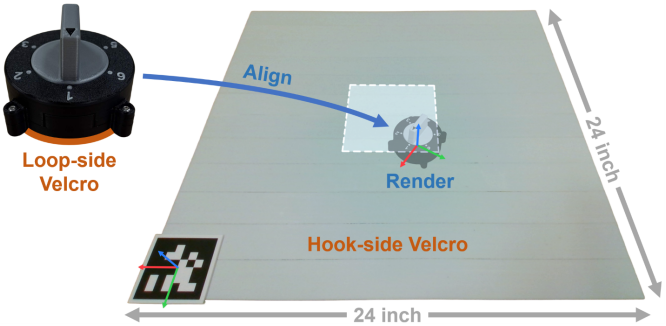


Fig. 10: We systematically sample task object initial positions within a specified region and render them in a 3rd-person camera view to guide the human operator in task object placement. When necessary, the task objects are attached to the workspace with Velcro.

Evaluation protocol. To generate task initial conditions systematically and make them easy to reproduce across research labs, we sample each object's initial pose at random and let the human operator place the object at that location. To hold objects at their sampled poses, we use a 24 inch by 24 inch acrylic board covered with hook-side Velcro. We apply loop-side Velcro to task-relevant objects (assembly block with screws, outlet, door, storage bin with utensils, platform) so they can be fixed at any location on the board. This method of fixing objects in the workspace also lets a single robot

% of episodes reaching	**DP** *vision*	**RDP** *+ tactile*	**$\pi_{0.5}$** *vision*	**FTP-1** *+ tactile*
Thread nut onto screw				
– initial contact	67	73	97	93
– nut balanced on screw	37	13	63	63
– threads engaged	27	10	57	50
– task terminated correctly	27	0	47	47
Tighten screw with Allen key				
– initial contact	97	80	93	97
– Allen key tip inserted	83	53	90	57
– screw driven by Allen key	83	47	90	53
– screw tightened	27	0	80	10
– task terminated correctly	0	0	57	10
Plug in charger				
– initial contact	100	97	100	100
– pins aligned	13	13	37	13
– fully inserted	7	13	20	13
– task terminated correctly	7	13	20	13
Turn doorknob to open door				
– knob grasped	100	100	100	100
– knob sufficiently rotated	100	97	100	97
– door opened	100	97	100	97
Retrieve single utensil from bin				
– initial contact	97		100	
– at least one utensil retrieved	53	*not evaluated*	57	*not evaluated*
– task terminated correctly with only one utensil on table	0		53	
Pick up card from platform				
– initial contact	100		57	
– card slid to edge	93	*not evaluated*	40	*not evaluated*
– card lifted by pivoting	57		10	
– card grasped	47		7	

Fig. 11: Task progress achieved by each policy, averaged over 30 evaluation trials.

arm perform tasks that would otherwise require a second arm to stabilize the object. The pose of each object's origin is sampled from a uniform distribution within a 0.15m by 0.15m area centered on the board. To provide visual guidance to the operator placing the objects, we render objects at their sampled poses with PyBullet and overlay them onto the board in a 3rd-person camera view. This procedure, shown in Figure 10 and similar to the setups in [56] and [15], is used for both demonstration collection and rollouts. For each task, we collect 100 demonstrations and evaluate 30 rollouts per policy, with robot initial poses during evaluation sampled from a Gaussian distribution fit to the UMI gripper initial poses in the demonstrations; the same 30 object and robot initial poses are used across all four policies for fair comparison. Each rollout has a one-minute time limit and is evaluated against the task's success criteria. For the six tasks in this section, we additionally measure task progress as the fraction of task-specific checkpoints completed.

B. What Are the Major Challenges for Vision-Only Policies on Contact-Rich Tasks?

We discuss the performance of the vision-only policies, Diffusion Policy (DP) and $\pi_{0.5}$, in this subsection. Excluding *turn doorknob to open door*, where both policies were able to solve the task with a 100% success rate, policy performance varies substantially across tasks. In the remaining tasks, except for *pick up card from platform*, we observe that $\pi_{0.5}$ outperforms DP by a significant margin, likely due to it having a more powerful model as well as a large and diverse pretraining data mix. *Plug in charger* appears to be the most difficult task for both policies, where $\pi_{0.5}$ leads with only 20% success rate. The rollouts we examined suggest that the low performance of vision-only policies across different tasks traces to several recurring challenges tied to the policies' inability to perceive or react to contact events. We describe these challenges below, with references to how each manifests across DP and $\pi_{0.5}$ rollouts on the six selected tasks. Figure 12 shows the challenges and their typical examples across tasks.

Establishing and maintaining contact. Every manipulation process requires contact to be established or maintained between the robot fingers and the target object, the target object and the environment, or both. However, vision-only policies frequently fail to sustain such required contact. In *thread nut onto screw*, policies sometimes miss the contact between the gripper finger and the nut when trying to roll it, leading to more attempts to properly engage the threads and longer task completion time. In *tighten screw with Allen key*, the Allen key can slip in-hand naturally over the course of tightening or when the tip becomes misaligned with the screw socket; in either case, it loses the contact needed to transmit torque when the robot fails to adjust its action to account for the shifted Allen key in-hand pose. In *retrieve single utensil from bin*, the policies are prone to making grasping attempts inside the utensil bin without properly contacting the chopsticks, causing either task failure or significantly longer, on average $2\times$, task completion time compared to human demonstrations. In *pick up card from platform*, the robot often fails to establish or maintain contact with the card during either sliding the card on the platform or pivoting the card to lift it up, leading to policies being unable to complete the task.

Precise alignment. Several tasks require the robot to precisely align an in-hand object with a target feature for the subsequent contact-driven step to succeed. The tolerances involved in such tasks and the required alignment adjustment actions are usually visually subtle. For *thread nut onto screw*, alignment errors are the dominant cause of failure, as even small offsets can cause the nut to fall off before the threads engage. When attempting to insert the Allen key tip into the screw socket in *tighten screw with Allen key*, misalignment between them leads to the Allen key colliding with the screw head and slipping in hand, making it more difficult to engage the Allen key with the screw properly. For *plug in charger*, alignment errors between the charging block pins and the outlet holes again dominate the failure cases. During demonstrations, the human operator often relies on tactile feedback to help guide and adjust the various alignment processes, especially for *plug in charger* where the pins are occluded by the charging block body in the wrist camera view. Across these tasks, such small adjustment-after-contact behaviors were not observed in the policy rollouts, hence presumably to not have been learned when the policy input

includes the vision modality alone.

Avoiding unsafe forceful interactions. Unless required by the task, excessive contact forces during task execution are usually undesired and can cause task failure or safety concerns. Vision-only policies struggle to avoid forceful collisions with task objects and environments across multiple tasks we evaluated them on. In *thread nut onto screw*, the gripper fingers can collide with the base block, which leads to spikes in contact forces. In *plug in charger*, the policy kept pushing the charging block into the outlet even when its pins are misaligned with the outlet holes and the insertion is blocked. In *retrieve single utensil from bin* and *pick up card from platform*, the robot can collide with the utensil bin or the flat platform and generate excessive forces. As shown in Table III, many rollouts generated contact forces above 40N.

Recognizing task-critical states. Certain task-critical states can be ambiguous to the vision modality alone. In *plug in charger*, whether the charger has been fully inserted can be ambiguous from the wrist camera view. Therefore, the policies sometimes fail the task even when the charging block is properly aligned with the outlet due to stopping the insertion motion before the charging block pins make contact with the end of the outlet slot, leaving the charging block only partially inserted. In *retrieve single utensil from bin*, vision-only policies are prone to grasping onto multiple chopsticks instead of just one, lacking the capability to recognize the number of chopsticks currently in between gripper fingers.

We also noticed that in *tighten screw with Allen key*, the policies tend to keep repeating the tightening motion after the screw has already been tightened. Although there is a strong visual signal at the moment the screw becomes tightened (Allen key tilts in-hand), a tightened vs. untightened screw looks visually the same otherwise. Since neither DP nor $\pi_{0.5}$ has memories of task stages completed, both tend to continue trying to tighten the screw by repeating the lift Allen key $\rightarrow$ align with screw $\rightarrow$ turn Allen key steps as if the screw has not been tightened. In general, we observe that DP is more prone to getting stuck in a loop and not terminate task execution compared to $\pi_{0.5}$, as DP also gets stuck in a loop in *retrieve single utensil from bin*. We view lack of memory as a crucial challenge for many types of manipulation tasks and not unique to contact-rich manipulation.

C. Do Visual-Tactile Policies Address These Challenges?

Tactile sensing is widely considered important for contact-rich manipulation, and intuitively, effective integration of force or tactile sensing can address the challenges discussed in Section V-B. In this subsection, we examine whether Reactive Diffusion Policy (RDP) [7] and Foundation Tactile Policy (FTP-1) [12], two recent representative visual-tactile policy learning methods, address such challenges encountered by their vision-only base policies. We used the official implementations provided by RDP and FTP-1, which both natively support the GelSight Mini tactile sensor used in our experiments. Table III and Figure 11 compare RDP and FTP-1 with their vision-only base policies, DP and $\pi_{0.5}$, on the four tasks for which we collected tactile data. On these tasks, RDP and FTP-1 did not meaningfully improve policy success rate, average task progress, interaction force, or completion time. Across policy rollouts on the four tasks, we observed three aspects that limited the benefit of the added tactile input from the GelSight Mini tactile sensor, which we discuss below.

Sensor coverage limited the tactile information available to the policy. In some cases, the policy performance is suboptimal not because the policies fail to properly react to the tactile input, but because the sensor never captured the relevant contact. In *thread nut onto screw*, both vision-only and visual-tactile policies produced forceful collisions between the back of the gripper finger and the object. This region is outside the GelSight Mini's sensing area, which covers only the inner fingertip surface. As a result, the tactile input does not capture these collision events, and the visual-tactile policies have no additional information to avoid them compared to their vision-only counterparts. Tasks *retrieve single utensil from bin* and *pick up card from platform* illustrate the same limitation. Although vision-only policies performed poorly on both tasks, we could not collect visual-tactile demonstrations or train visual-tactile policies for them because GelSight Mini's fingertip-only coverage and bulky form factor made it unsuitable for either task.

Policies fail to learn the tactile-driven behaviors present in demonstrations. During data collection, the demonstrator relied on tactile feedback for behaviors such as establishing and maintaining contact and tactile-driven alignment, which vision-only policies struggled to reproduce as discussed in Section V-B. These signals are captured by tactile sensors and are used as input to RDP and FTP-1 during both training and inference. However, the two policies exhibited similar failures to their vision-only base policies, including improper engagement between the Allen key tip and the screw socket in *tighten screw with Allen key*, and misalignment between task objects in *thread nut onto screw*, *tighten screw with Allen key*, and *plug in charger*. This gap was most evident in *plug in charger*, where the charging block pins and outlet holes are visually occluded during insertion, and the demonstrator relied on tactile feedback to search for the holes and align the pins with them. As noted in Section V-B, the vision-only policies did not show this search behavior, and the same failure also occurred in the tactile-augmented policies despite receiving tactile signals that may have helped them avoid it.

Policies acted on incidental tactile signals rather than task-relevant ones. When the visual-tactile policies did react to tactile input, they sometimes reacted to signals that do not reliably indicate the current task state. In *tighten screw with Allen key*, both visual-tactile policies tended to stop the tightening motion prematurely when the Allen key slipped enough in the gripper, as shown in Figure 13. Such in-hand slip can occur naturally as the task progresses or from misalignment between the Allen key tip and the screw socket, hence it does not reliably indicate that the screw has been fully tightened. This premature termination contributed substantially to both visual-tactile policies significantly underperforming their vision-only base policies on this task.

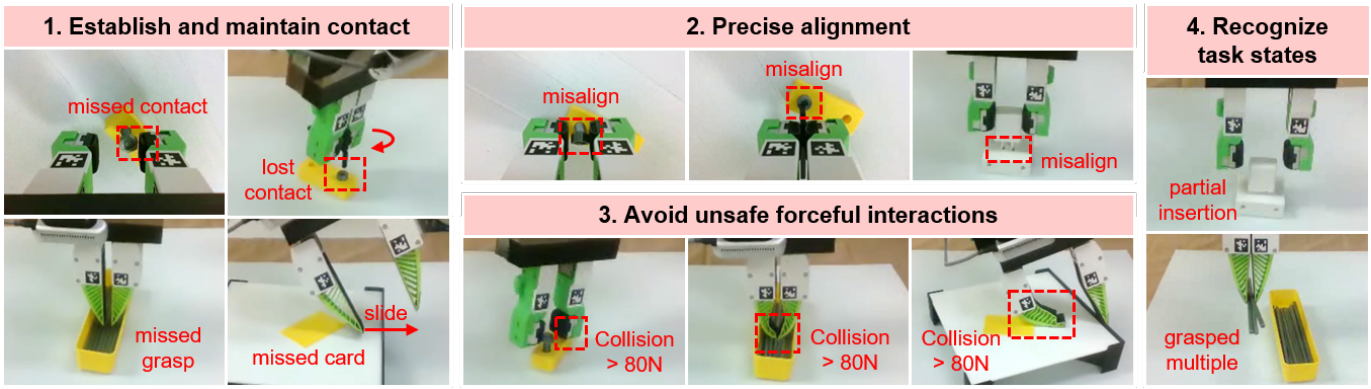


Fig. 12: Examples of the four typical challenges vision-only policies encountered on the selected tasks.

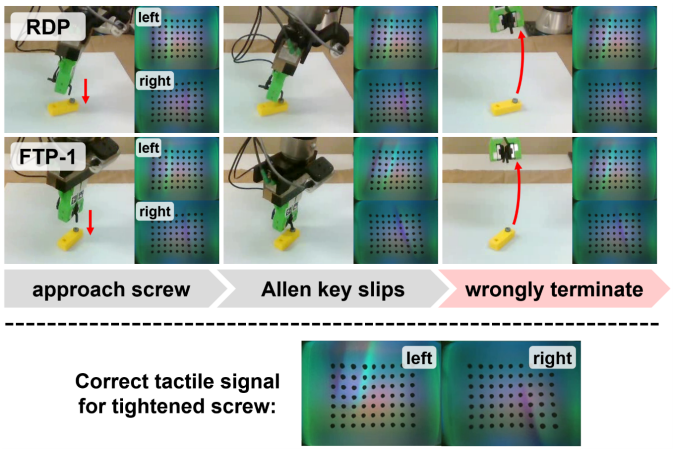


Fig. 13: Top: Both RDP and FTP-1 are prone to terminating the tightening motion without tightening the screw after the Allen key slips in hand. Bottom: Although in-hand slip can happen naturally as the task progresses, the correct tactile signal associated with a tightened screw has a large in-plane torque.

D. Case Studies on Behavioral Generalization

Section III-C introduces task variations designed to evaluate behavioral generalization capabilities of trained policies. Here we use *turn doorknob to open door* to conduct two case studies on behavioral generalization. We select this task since it is the only one for which all four policies achieve near-perfect success rates, making it the ideal candidate where a drop in success rate under a variation can be attributed to the variation rather than the base task difficulty. We first test a behavioral variation in the amount of rotation required to unlock the door. We use doors that require approximately 72-degree and 80-degree rotations to unlock, where the base door used in training requires 60 degrees. The 72-degree and 80-degree doors are visually indistinguishable from the base 60-degree door. We also test a visual + behavioral variation in the doorknob geometry, where the original doorknob is replaced with one that is 5mm smaller in diameter.

1) Behavioral variation: required rotation amount to unlock door: Figure 14 reports success rates for the four policies under the three rotation-amount conditions. While all four policies show degraded performance as the required rotation increases, the extent of degradation differs substantially. While the success rates of DP, $\pi_{0.5}$, and FTP-1 dropped by half for the 72-degree door and dropped to almost zero for the 80-degree door, RDP is able to partially retain its performance at 80% and 53.3% success rates.

To understand what is driving the performance gap between policies, we compute the actual amount by which each policy rotates the doorknob during rollouts. If a policy is adapting to the change in required rotation, its wrist rotation angle distribution should shift to match the increased required rotation amount; otherwise, its wrist rotation angle distribution would remain roughly unchanged, and successes would only occur when the executed rotation happens to exceed the requirement. Figure 14 shows the per-rollout wrist rotation for each policy across the three conditions. For DP, $\pi_{0.5}$, and FTP-1, their wrist rotation distributions remain largely unchanged across the base, 72-degree, and 80-degree conditions. This indicates that these policies rotate the doorknob by roughly the same amount regardless of how much rotation is required to unlock the door, and the successes that do occur under the variations correspond to rollouts where the natural spread of the wrist rotation distribution happens to reach the increased rotation requirement. RDP, by contrast, has its wrist rotation distribution shifted upward as the required rotation increases, indicating it is adapting to the task variations whereas the other three policies are not.

These results show that tactile-augmented policies can achieve better behavioral generalization even when they offer no gains in the in-distribution task settings. However, how tactile feedback is integrated matters. FTP-1, which also uses tactile sensing, does not exhibit RDP's adaptation behavior: its wrist rotation distribution remains flat across the three conditions, similar to the vision-only policies. Thus, for the task variations we tested here for *turn doorknob to open door*, incorporating tactile feedback alone is insufficient, and RDP's integration mechanism appears necessary.

2) Visual + behavioral variation: smaller doorknob diameter: We replace the base 60mm doorknob with one 5mm smaller in diameter, subtly altering both the doorknob's appearance and the grasping width required to hold it. Across 10 rollouts per policy, all four policies achieved 0% success rate.

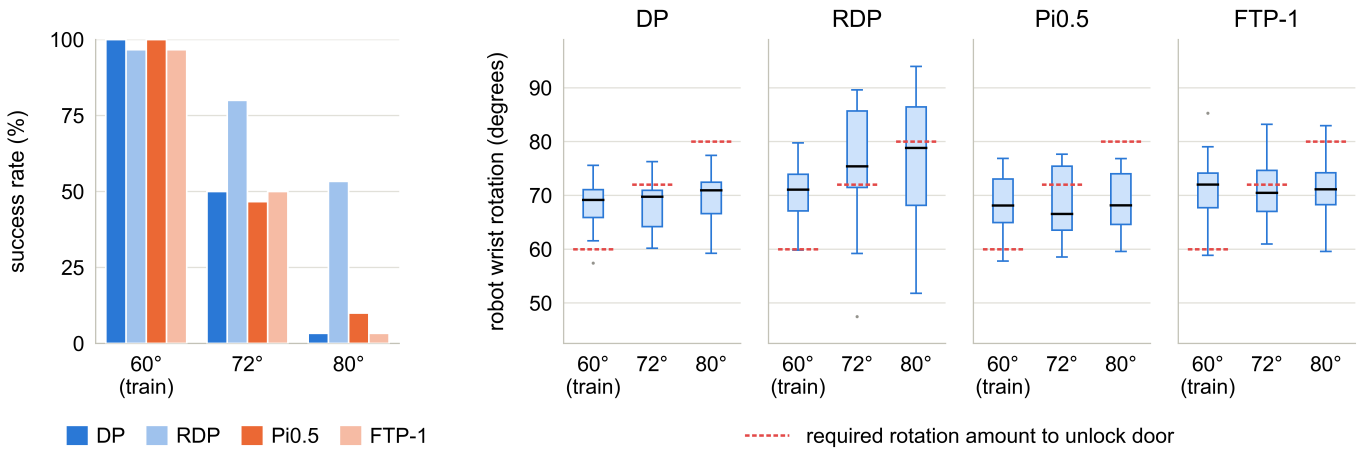


Fig. 14: Left: Success rates across task variants for *turn doorknob to open door*. Right: Robot wrist rotation angle distribution for each policy. Only RDP shows behavior adaptation effort when the task demands larger rotation angles to unlock the door.

TABLE IV: Gripper width reached (mm) while trying to turn the doorknob, reported as mean ± std across episodes (30 episodes for the 60 mm knob, 10 for the 55 mm knob).

Variant	DP	RDP	$\pi_{0.5}$	FTP-1
60 mm knob (train)	58.9 ± 0.12	59.0 ± 0.13	58.8 ± 0.13	59.2 ± 0.14
55 mm knob	59.0 ± 0.11	59.0 ± 0.11	58.7 ± 0.11	58.9 ± 0.14

We further compute the gripper width each policy converged at for gripping the doorknob in Table IV, which shows that all policies attempted to grip the smaller 55mm doorknob with the same gripper width used for the 60mm doorknob in the base configuration the policy was trained on. This causes the robot to miss the smaller doorknob entirely and fail to establish a grasp. Because the failure was immediate and consistent across every rollout of every policy, we did not run the full 30-rollout protocol on this variation. This consistent failure to adjust the grasping width indicates that all four policies failed to learn a key aspect of grasping: adapting to the object's size to achieve stable contact.

VI. Discussion

As a first release, ContactBench 1.0 makes several design choices that shape what it can and cannot evaluate. Here we discuss its coverage of tasks and materials, the limitations introduced by 3D-printed assets, and the broader evaluation of robot systems it is intended to support.

Task and material coverage. The initial release of ContactBench 1.0 focuses on rigid and articulated objects. This scope reflects the time and resources available for the current study and captures only part of the diversity of everyday contact-rich manipulation. We envision the benchmark as a growing community effort rather than a fixed collection of tasks. Future extensions can increase the number of tasks and assets and introduce broader variations within existing tasks, enabling more comprehensive evaluation of skill acquisition and generalization. Beyond rigid and articulated objects, an important direction could be to incorporate different forms of deformable objects, fluids, and granular materials. These extensions will require corresponding physical assets, simulation capabilities, and evaluation protocols that account for the distinct manipulation challenges of these materials.

Surface properties of printed assets. Although 3D printing facilitates reproducible fabrication, it also introduces surface characteristics that may differ from those of everyday manufactured objects. In particular, layer-induced surface texture can produce dense tactile features that are observable by sensors such as GelSight. This limitation concerns the transferability of learned policies beyond the benchmark assets. Furthermore, surface friction is not systematically characterized or varied in the current benchmark. Future work can investigate new formats of reproducible assets that can better leverage variance in surface texture and properties.

What evaluation will ContactBench 1.0 enable? ContactBench 1.0 is designed to evaluate robotic systems across a broad range of contact-rich manipulation tasks. Our experiments illustrate its use in evaluating manipulation policies, but performance also depends on other aspects of a robotic system, including mechanical design, embodiment selection, and sensor design, placement, and coverage. The benchmark can therefore support comparisons across these dimensions, helping researchers investigate how hardware and sensing choices affect manipulation performance. We envision ContactBench 1.0 as a shared experimental resource for evaluating complete robotic systems and informing the joint design of hardware, sensing, and control.

VII. Conclusion

We presented ContactBench 1.0, a benchmark for contact-rich robotic manipulation grounded in observations of everyday human activities. Organized around eight core manipulation skills, the benchmark provides 36 reference tasks and more than 40 standardized, 3D-printable assets with corresponding simulation models. Experiments across different 3D printers and comparisons between physical and

simulated assets validate fabrication reproducibility and the correspondence of task-relevant mechanical responses, establishing a foundation for reproducible experimentation across laboratories. Beyond asset models, we provide a simulation suite integrating the benchmark assets, a GelSight tactile simulator, a teleoperation interface for data collection, and scripted policies. Evaluating representative vision-only and visual-tactile policies on six benchmark tasks, we find that contact-rich manipulation remains challenging for the policy learning methods we tested. We hope ContactBench 1.0 will serve as a shared testbed for developing, evaluating, and comparing robot systems that address these challenges.

REFERENCES

[1] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[2] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-rl for insertion: Generalization to objects of unknown geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6437–6443.

[3] S. Kim and A. Rodriguez, "Active extrinsic contact sensing: Application to general peg-in-hole insertion," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 241–10 247.

[4] C. Higuera, J. Ortiz, H. Qi, L. Pineda, B. Boots, and M. Mukadam, "Perceiving extrinsic contacts from touch improves learning insertion policies," *arXiv preprint arXiv:2309.16652*, 2023.

[5] P. Jin, B. Huang, W. W. Lee, T. Li, and W. Yang, "Visual-force-tactile fusion for gentle intricate insertion tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4830–4837, 2024.

[6] Y. Delgado, J. G. Buenaventura-Carreón, F. Erich, R. Mykhailyshyn, T. Motoda, K. Makihara, and Y. Domae, "Peg-in-bench: A modular benchmark for high-precision robotic insertion," 2026. [Online]. Available: https://arxiv.org/abs/2609.00906

[7] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu, "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," *arXiv preprint arXiv:2503.02881*, 2025.

[8] W. Liu, J. Wang, Y. Wang, W. Wang, and C. Lu, "ForceMimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 1105–1112.

[9] Z. He, H. Fang, J. Chen, H.-S. Fang, and C. Lu, "FoAR: Force-aware reactive policy for contact-rich robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5625–5632, 2025.

[10] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[11] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, b. ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 17–40. [Online]. Available: https://proceedings.mlr.press/v305/black25a.html

[12] C. Yuan, Z. Zhang, M. Zhou, W. Chen, Y. Wang, Z. Liu, D. Niu, S. Wang, H. Zhang, W. Zhang *et al.*, "Ftp-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation," *arXiv preprint arXiv:2606.13102*, 2026.

[13] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[14] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji, "Benchmarking protocols for evaluating small parts robotic assembly systems," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 883–889, 2020.

[15] Y. Chen, K. Kimble, E. H. Adelson, T. Asfour, P. Chanrungmaneekul, S. Chitta, Y. Chitambar, Z. Chen, K. Goldberg, D. Kragic *et al.*, "Manipulationnet: An infrastructure for benchmarking real-world robot manipulation with physical skill challenges and embodied multimodal reasoning," *arXiv preprint arXiv:2603.04363*, 2026.

[16] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1863–1891, 2025.

[17] Y. Tian, J. Jacob, Y. Huang, J. Zhao, E. L. Gu, P. Ma, A. Zhang, F. Javid, B. Romero, S. Chitta, S. Sueda, H. Li, and W. Matusik, "Fabrica: Dual-arm assembly of general multi-part objects via integrated planning and learning," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 406–419. [Online]. Available: https://proceedings.mlr.press/v305/tian25a.html

[18] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, "Fmb: a functional manipulation benchmark for generalizable robotic learning," *The International Journal of Robotics Research*, vol. 44, no. 4, pp. 592–606, 2025.

[19] D. Liconti, Y. Zhou, Y. Toshimitsu, R. Hinchet, A. X. Appius, and R. K. Katzschmann, "A benchmark of dexterity for anthropomorphic robotic hands," *IEEE Robotics and Automation Letters*, 2026.

[20] Y. Wang, B. Tang, C. Gan, D. Fox, K. Mo, Y. Narang, and I. Akinola, "Matchmaker: Automated asset generation for robotic assembly," *arXiv preprint arXiv:2503.05887*, 2025.

[21] D. Fortier, T. Adamson, T. Hellebrekers, T. LaScala, K. Ennin, M. Murray, A. Kolobov, and G. Mullins, "Benchmarking affordance generalization with busybox," 2026. [Online]. Available: https://arxiv.org/abs/2602.05441

[22] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[23] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[24] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2025.

[25] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.

[26] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *arXiv preprint arXiv:2410.00425*, 2024.

[27] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327–7334, 2022.

[28] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[29] Y. Li, X. Zhang, R. Wu, Z. Zhang, Y. Geng, H. Dong, and Z. He, "Unidoormanip: Learning universal door manipulation policy over large-scale and diverse door manipulation environments," *arXiv preprint arXiv:2403.02604*, 2024.

[30] Z. Si and W. Yuan, "Taxim: An example-based simulation model for gelsight tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2361–2368, 2022.

[31] Y. Lin, J. Lloyd, A. Church, and N. F. Lepora, "Tactile gym 2.0: Sim-to-real deep reinforcement learning for comparing low-cost high-resolution robot touch," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, p. 10754–10761, Oct. 2022. [Online]. Available: http://dx.doi.org/10.1109/LRA.2022.3195195

[32] E. Miller, J. Reddy, A. Deshmukh, T. McInroe, D. Abel, O. M. Aodha, and S. Vijayakumar, "roto 2.0: The robot tactile olympiad," 2026. [Online]. Available: https://arxiv.org/abs/2605.21429

[33] Q. K. Luu, P. Zhou, Z. Xu, Z. Zhang, Q. Qiu, and Y. She, "Manifeel: Benchmarking and understanding visuotactile manipulation policy learning," *arXiv preprint arXiv:2505.18472*, 2025.

[34] B. Chen, W. Wan, T. Chen, X. Guo, C. Xu, Y. Qi, H. Zhang, L. Wu, T. Xu, Z. Li *et al.*, "Univtac: A unified simulation platform for visuo-tactile manipulation data generation, learning, and benchmarking," *arXiv*

preprint arXiv:2602.10093, 2026.
[35] Y. Wu, Q. Li, N. Min, G. Zhong, H. Zhao, Z. Li, H. Xu, S. Xu, X. Lin, Z. Diao *et al.*, "Openvitac: Learning and benchmarking visuo-tactile policies in a unified sim-and-real framework," *arXiv preprint arXiv:2610.10384*, 2026.
[36] R. Zheng, D. Niu, Y. Xie, J. Wang, M. Xu, Y. Jiang, F. Castañeda, F. Hu, Y. L. Tan, L. Fu *et al.*, "Egoscale: Scaling dexterous manipulation with diverse egocentric human data," *arXiv preprint arXiv:2602.16710*, 2026.
[37] H. Luo, Y. Feng, W. Zhang, S. Zheng, Y. Wang, H. Yuan, J. Liu, C. Xu, Q. Jin, and Z. Lu, "Being-h0: vision-language-action pretraining from large-scale human videos," *arXiv preprint arXiv:2507.15597*, 2025.
[38] T. Tao, M. K. Srirama, J. J. Liu, K. Shaw, and D. Pathak, "Dexwild: Dexterous human interactions for in-the-wild robot policies," *arXiv preprint arXiv:2505.07813*, 2025.
[39] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2016.
[40] X. Zhu, B. Huang, and Y. Li, "Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper," *Advances in Neural Information Processing Systems*, vol. 38, pp. 153 783–153 812, 2025.
[41] A. Bronars, S. Kim, P. Patre, and A. Rodriguez, "Texterity: Tactile extrinsic dexterity," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 7976–7983.
[42] J. Toskov, R. Newbury, M. Mukadam, D. Kulic, and A. Cosgun, "In-hand gravitational pivoting using tactile sensing," in *Conference on Robot Learning*. PMLR, 2023, pp. 2284–2293.
[43] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang, "Robot synesthesia: In-hand manipulation with visuotactile sensing," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6558–6565.
[44] M. Yang, C. Lu, A. Church, Y. Lin, C. Ford, H. Li, E. Psomopoulou, D. A. Barton, and N. F. Lepora, "Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch," *arXiv preprint arXiv:2405.07391*, 2024.
[45] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning*. PMLR, 2023, pp. 2549–2564.
[46] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, "In-hand object rotation via rapid motor adaptation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1722–1732.
[47] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand reorientation of novel and complex object shapes," *Science Robotics*, vol. 8, no. 84, p. eadc9244, 2023.
[48] J. Wang, Y. Yuan, H. Che, H. Qi, Y. Ma, J. Malik, and X. Wang, "Lessons from learning to spin "pens"," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 3124–3138. [Online]. Available: https://proceedings.mlr.press/v270/wang25j.html
[49] J. Gao, S. Belkhale, S. Dasari, A. Balakrishna, D. Shah, and D. Sadigh, "A taxonomy for evaluating generalist robot manipulation policies," *IEEE Robotics and Automation Letters*, vol. 11, no. 3, pp. 3182–3189, 2026.
[50] The Newton Contributors, "Newton: GPU-accelerated physics simulation for robotics and simulation research," Apr. 2025. [Online]. Available: https://github.com/newton-physics/newton
[51] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
[52] Google DeepMind and NVIDIA, "MuJoCo Warp," https://github.com/google-deepmind/mujoco_warp, 2025.
[53] R. Elandt, E. Drumwright, M. Sherman, and A. Ruina, "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 8238–8245.
[54] 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.
[55] Valve Software, "OpenVR SDK," https://github.com/ValveSoftware/openvr, 2026.
[56] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina *et al.*, "A careful examination of large behavior models for multitask dexterous manipulation," *arXiv preprint arXiv:2507.05331*, 2025.

APPENDIX A
SIMULATION DETAILS

A. Asset modeling

Fig. A1 shows how the modeling choices of Section IV apply to four representative mechanisms; in each case, the model preserves the task-relevant contact behavior and motion constraints rather than the exact printed geometry. The *push-and-turn dial* pairs an axially compliant joint with a pin-and-post gate that approximates the small printed blocking features on the real part. Pressing the dial moves the pin downward, clear of the posts, so the dial can rotate; releasing it at the home orientation lets the spring push it back up. The *door handle* retracts a sliding latch when turned and returns to its rest position through a revolute spring–damper. The *key and keyway* uses axial sliding joint represents key insertion. The *door knob* similarly converts spindle rotation into latch retraction, with contact between the latch and strike plate keeping the door shut until the latch clears. For these three door mechanisms, we use a reduced transmission model that maps spindle rotation to rack displacement and reflects any load on the rack back to the spindle as torque.

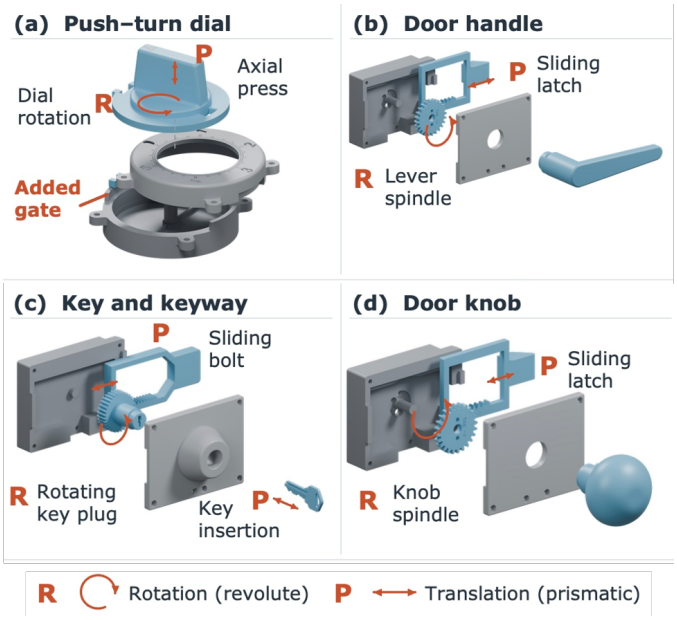


Fig. A1: **Examples of CAD-derived simulation assets.** The assets approximate the corresponding physical mechanisms using rigid parts, joints, and simplified representations of compliance and mechanical constraints. Exploded views of **(a)** a push-and-turn dial, **(b)** a door handle, **(c)** a key-operated lock, **(d)** a doorknob, and **R** and **P** denote revolute and prismatic degrees of freedom, respectively.

B. Tactile sensing simulation

This section gives the formulation of the tactile sensing module summarized in Section IV. Within the hydroelastic pressure-field framework [53], the gel follows the pressure–indentation relation

$$p_{\text{law}}(\delta) = k_n \delta \left[1 + \alpha \left(\frac{\delta}{h}\right)^2\right], \tag{A1}$$

where $\delta \geq 0$ is local indentation, h is gel thickness, k_n is the normal foundation stiffness, and $\alpha \geq 0$ controls stiffening under compression. Setting $\alpha = 0$ recovers a linear response.

The contact surface is rasterized into a normal-pressure map $p_0(\mathbf{x})$. For nonzero contact, this map is normalized as

$$\widehat{p}(\mathbf{x}) = \frac{F_n^{\text{sim}}}{\int_\Omega p_0(\mathbf{x})\,\mathrm{d}A} p_0(\mathbf{x}), \tag{A2}$$

where Ω is the sensing domain and F_n^{sim} is the solver-resolved normal load on the gel pad. Pressure exceeding the full-compression level is redistributed to neighboring cells, yielding the pressure map p used for subsequent deformation computation.

At each contacting surface location, tangential displacement is updated by

$$\begin{aligned} \mathbf{u}_{t+\Delta t} &= \Pi_{\|\mathbf{u}\| \leq u_{\text{lim}}} \left(\mathbf{u}_t + \mathbf{v}_\parallel \Delta t\right), \\ u_{\text{lim}} &= \min\left(\frac{\mu p}{k_s}, u_{\text{cap}}\right), \end{aligned} \tag{A3}$$

where $\mathbf{v}_\parallel$ is the local relative tangential velocity, k_s is the shear foundation stiffness, μ is the friction coefficient, and u_{cap} limits shear displacement. The operator Π projects onto the specified displacement bound, ensuring that the shear traction $\boldsymbol{\tau} = k_s \mathbf{u}$ satisfies $\|\boldsymbol{\tau}\| \leq \mu p$.

The resulting deformation fields are rendered into tactile images with the Taxim-based optical shading [30] described in Section IV. Together, the module's outputs expose contact geometry and surface texture, distributed force and torque, incipient slip, and the aggregate wrench on each finger.

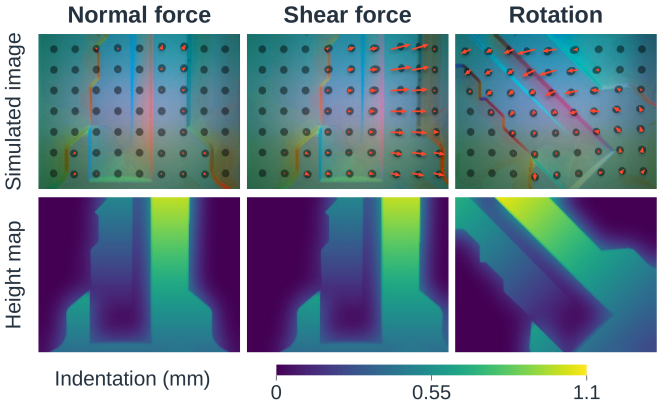


Fig. A2: Simulated GelSight Mini responses to a key under normal loading, shearing, and rotation. Top: rendered tactile images, with red arrows indicating marker displacement. Bottom: gel indentation height maps on a common color scale.

Figure A2 compares the simulated sensor responses to normal loading, tangential shearing, and rotation of a key, shown from left to right. The top row shows rendered tactile images, with red arrows indicating the direction and relative magnitude of marker displacement. The bottom row shows the corresponding height maps of gel indentation in millimeters, using a common color scale across all three conditions.

APPENDIX B
POLICY TRAINING DETAILS

We use the official implementations of the four policy learning methods we evaluated in Section V:

- Diffusion Policy (commit 5ba07ac)
- Reactive Diffusion Policy (commit 8ed82a3)
- $\pi_{0.5}$ (commit 15a9616)
- Foundation Tactile Policy (commit 89fa681)

We show our real-world setup in Fig. B1, policy training details in Table B1, and demonstration statistics and the language instructions used for $\pi_{0.5}$ and FTP-1 in Table B2.

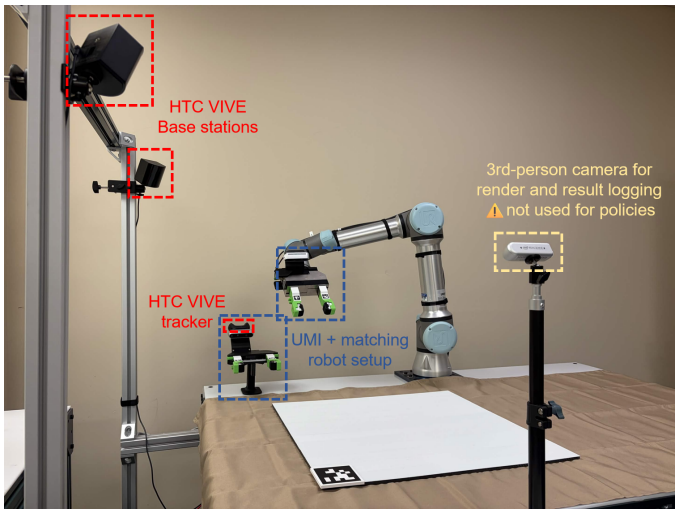


Fig. B1: Our real-world hardware setup for demonstration collection and policy rollout.

TABLE B1: Training details of the evaluated policies.

	DP	**RDP**	**$\pi_{0.5}$**	**FTP-1**
Initialization	From scratch		`pi05_base`, full fine-tuning	`ftp1_v0426_50kstep`, full fine-tuning
Data rate	12 Hz	12 Hz (slow policy), 24 Hz (fast policy)	24 Hz	
Visual input	Wrist RGB, 240 × 320		Wrist RGB, 240 × 320, resized then zero-padded to 224 × 224	
Tactile input	–	GelSight Mini marker displacements from both fingers, each projected to 15-D with the PCA basis released with RDP	–	GelSight Mini images from both fingers, resized to 224 × 224; tactile encoder initialized from FTP-1's pretrained GelSight Mini checkpoint
Proprioceptive input	EEF pose (relative to current), absolute gripper width (normalized)		Absolute gripper width (normalized); EEF pose set to identity	Absolute gripper width; EEF pose set to identity
Observation history	2 steps		1 step (current only)	
Action	10-D: position (3) and 6D rotation (6) relative to the current EEF pose, absolute gripper width (1)			
Prediction horizon	16 steps (1.33 s)		32 steps (1.33 s)	
Batch size	64			
Peak LR / schedule	1×10^{-4}; cosine decay, 500 warmup steps	AT: 1×10^{-3}, cosine decay, 100 warmup steps; LDP: 1×10^{-4}, cosine decay, 500 warmup steps	5×10^{-5}; cosine decay to 5×10^{-6}, 500 warmup steps	
Training length	501 epochs	AT: 601 epochs; LDP: 501 epochs	40k steps for *thread nut onto screw* and *tighten screw with Allen key*, 20k for the rest	
Hardware	1× RTX 5090		2–4× GH200	2× GH200

EEF: end effector. AT: asymmetric tokenizer (RDP's fast policy). LDP: latent diffusion policy (RDP's slow policy).

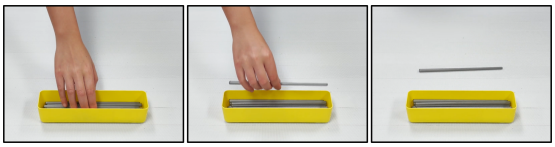
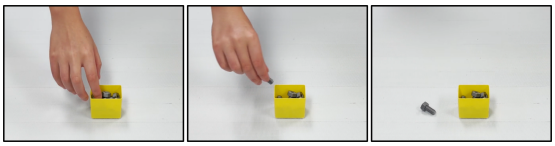
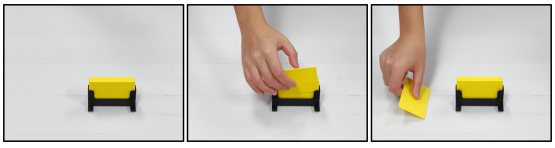
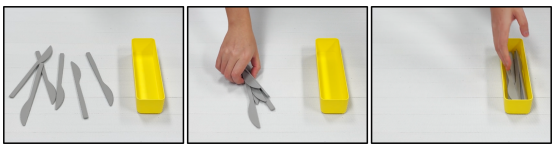
TABLE B2: Demonstrations used for training and language instructions given to $\pi_{0.5}$ and FTP-1.

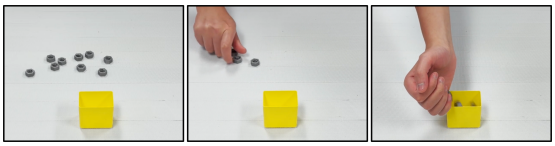
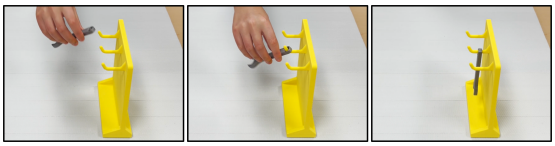
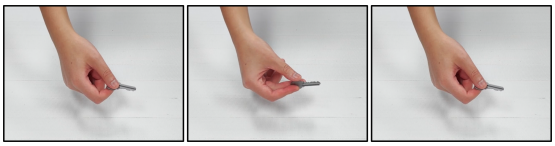
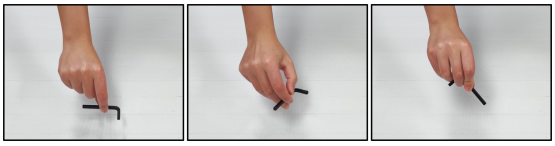
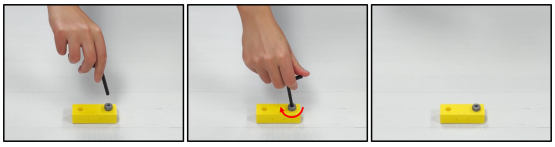
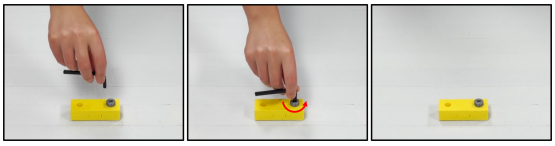
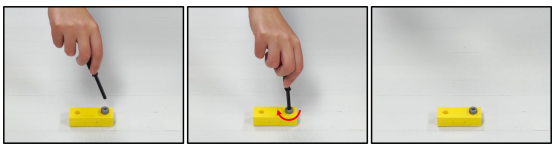
Task	Demos	Mean (s)	Min–max (s)	Language instruction
Thread nut onto screw	100	19.6	14.0–24.0	"screw the nut onto the screw"
Tighten screw with Allen key	100	21.0	11.9–32.7	"tighten the screw with allen key"
Plug in charger	100	8.4	7.0–10.0	"plug the charger into the outlet"
Turn doorknob to open door	99*	8.8	7.5–11.0	"turning the door knob to open the door"
Retrieve single utensil from bin	100	12.9	8.1–27.5	"pick up a single utensil from the storage bin"
Pick up card from platform	100	8.6	7.0–10.5	"pick up the thin object on the table"

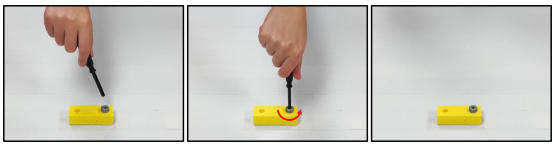
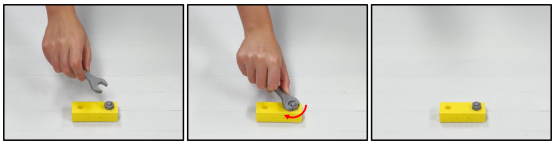
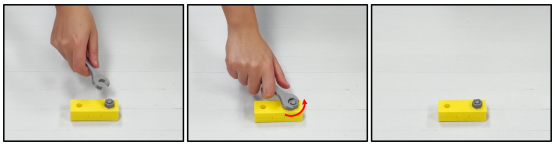
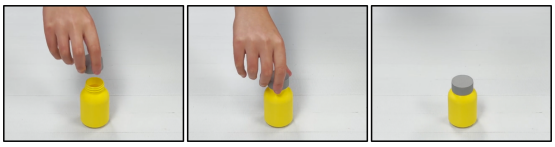
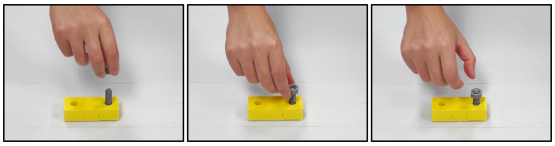
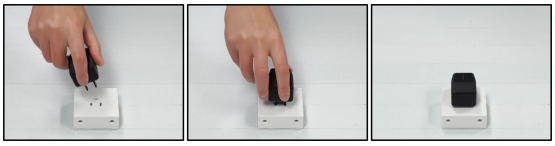
All demonstrations were recorded at 24 Hz with the UMI gripper. *One of the 100 recorded demonstrations was dropped due to not passing data quality checks for the extracted UMI gripper widths. FTP-1 uses the same instructions with a terminating period.

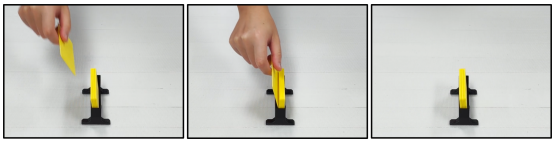
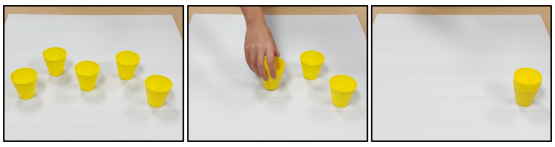
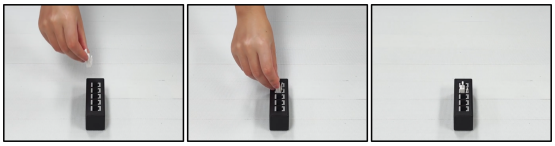
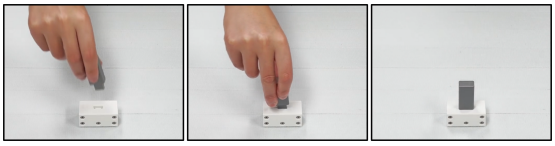
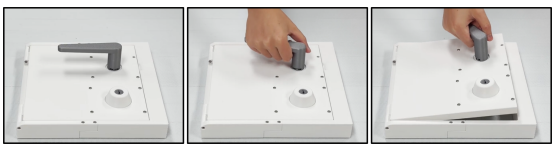
APPENDIX C
DETAILED BENCHMARK TASK DESCRIPTIONS

TABLE C1: Base benchmark tasks

Task Execution Sequence	Task Specs
	**Task:** Grasp target key from keychain **Skills:** Grasping from clutter, grasping thin or small objects **Description:** For a keychain in the workspace with multiple keys, find and singulate the specified target key among all keys on the keychain. **Success criteria:** The robot movement stops with only the specified target key stably grasped within the robot end-effector.
	**Task:** Retrieve single utensil from bin **Skills:** Grasping from clutter, grasping thin or small objects **Description:** A utensil storage bin is placed in the workspace. The target utensil is placed in the bin alongside multiple other utensils. Pick up a single copy of the target utensil from the cluttered storage bin and place it on the table without removing other objects from the bin. **Success criteria:** The robot movement stops after separating a single copy of the target utensil and placing it on the table. Other utensils should remain in the storage bin. The robot should not remain in contact with the task objects.
	**Task:** Retrieve single fastener from bin **Skills:** Grasping from clutter, grasping thin or small objects **Description:** A fastener storage bin is placed in the workspace. The target fastener is placed in the bin alongside multiple other fasteners. Pick up a single copy of the target fastener from the cluttered storage bin and place it on the table without removing other objects from the bin. **Success criteria:** The robot movement stops after separating a single copy of the target fastener and placing it on the table. Other fasteners should remain in the storage bin. The robot should not remain in contact with the task objects.
	**Task:** Retrieve top cup from stack **Skills:** Grasping from clutter **Description:** A neatly organized stack of cups is placed in the workspace. Pick up the top cup from the stack and place it on the table without disturbing the rest of the stack. **Success criteria:** The robot movement stops after separating the top cup from the stack and placing it on the table. The remaining cups should remain in a neat stack. The robot should not remain in contact with the task objects.
	**Task:** Retrieve single card from deck **Skills:** Grasping from clutter, grasping thin or small objects **Description:** A neatly organized deck of cards is placed in a cardholder in the workspace. Pick up a single card and place it on the table without disrupting the rest of the card deck. **Success criteria:** The robot movement stops after separating a single card from the deck and placing it on the table. The remaining cards should remain a neat deck in the cardholder. The robot should not remain in contact with the task objects.
	**Task:** Pick up object from platform **Skills:** Grasping thin or small objects **Description:** Pick up the thin and/or small object placed on the flat platform. **Success criteria:** The target object is stably grasped by the robot and lifted off the platform.
	**Task:** Collect utensils **Skills:** Grasping multiple objects, placing under environmental constraints **Description:** Collect all utensils scattered across the workspace and place them into the storage bin. Being able to hold multiple utensils simultaneously will greatly increase task completion efficiency. **Success criteria:** The robot movement stops after placing all utensils inside the storage bin. The robot should not remain in contact with the task objects.

Task Execution Sequence	Task Specs
	**Task:** Collect fasteners **Skills:** Grasping multiple objects, placing under environmental constraints **Description:** Collect all fasteners scattered across the workspace and place them into the storage bin. Being able to hold multiple fasteners simultaneously will greatly increase task completion efficiency. **Success criteria:** The robot movement stops after placing all fasteners inside the storage bin. The robot should not remain in contact with the task objects.
	**Task:** Hang object **Skills:** Placing under environmental constraints **Description:** Hang the object in-hand onto an empty hanging board hook. **Success criteria:** The robot movement stops after placing the target object onto a hanging hook. The robot should not remain in contact with the task objects.
	**Task:** Place utensil into bin **Skills:** Placing under environmental constraints **Description:** A utensil bin with utensils already stored in it is placed in the workspace. Starting with a utensil held in hand, find an open pocket in the cluttered utensil storage bin and place the held utensil in it. **Success criteria:** The robot movement stops after placing the target object into the storage bin. The robot should not remain in contact with the task objects.
	**Task:** In-hand repose **Skills:** In-hand repositioning **Description:** Manipulate the target object in-hand, such that it ends up in the desired pose relative to the hand. Examples include flipping the orientation of a key or a USB stick to align with the slot. **Success criteria:** The target object pose converges and stabilizes to the desired pose, and remains within the robot end-effector throughout the execution without being dropped.
	**Task:** In-hand grasp transition **Skills:** In-hand repositioning **Description:** Manipulate the target object in-hand, such that it ends up with the desired grasp type (see grasp types defined in the Grasp Taxonomy [39]). **Success criteria:** The target object pose converges and stabilizes with the desired grasp type, and remains within the robot end-effector throughout the execution without being dropped.
	**Task:** Tighten screw with Allen key **Skills:** Extrinsic contact control, insertion and alignment **Description:** A screw is placed into the screw slot of an assembly block in the workspace, but the threads are unengaged. Use the Allen key grasped in hand to tighten the screw. **Success criteria:** The screw is tightened and remains stationary when a small force is applied in an attempt to rotate the screw. The robot movement stops after terminating the tightening motion, moving away from, and breaking contact with the screw.
	**Task:** Loosen screw with Allen key **Skills:** Extrinsic contact control, insertion and alignment **Description:** A screw is placed into the screw slot of an assembly block in the workspace and is fully tightened. Use the Allen key grasped in hand to loosen the screw. **Success criteria:** The screw is loosened and can rotate freely when a small force is applied in an attempt to rotate the screw. The robot movement stops after terminating the loosening motion, moving away from, and breaking contact with the screw.
	**Task:** Tighten screw with screwdriver **Skills:** Extrinsic contact control, insertion and alignment **Description:** A screw is placed into the screw slot of an assembly block in the workspace, but the threads are unengaged. Use the screwdriver grasped in hand to tighten the screw. **Success criteria:** The screw is tightened and remains stationary when a small force is applied in an attempt to rotate the screw. The robot movement stops after terminating the tightening motion, moving away from, and breaking contact with the screw.

Task Execution Sequence	Task Specs
	**Task:** Loosen screw with screwdriver **Skills:** Extrinsic contact control, insertion and alignment **Description:** A screw is placed into the screw slot of an assembly block in the workspace and is fully tightened. Use the screwdriver grasped in hand to loosen the screw. **Success criteria:** The screw is loosened and can rotate freely when a small force is applied in an attempt to rotate the screw. The robot movement stops after terminating the loosening motion, moving away from, and breaking contact with the screw.
	**Task:** Tighten nut with wrench **Skills:** Extrinsic contact control, insertion and alignment **Description:** A nut is threaded onto a screw head in the workspace but is left untightened. Use the wrench grasped in hand to tighten the nut. **Success criteria:** The nut is tightened and remains stationary when a small force is applied in an attempt to rotate the nut. The robot movement stops after terminating the tightening motion, moving away from, and breaking contact with the nut.
	**Task:** Loosen nut with wrench **Skills:** Extrinsic contact control, insertion and alignment **Description:** A nut is threaded onto a screw head in the workspace and is fully tightened. Use the wrench grasped in hand to loosen the nut. **Success criteria:** The nut is loosened and can rotate freely when a small force is applied in an attempt to rotate the nut. The robot movement stops after terminating the loosening motion, moving away from, and breaking contact with the nut.
	**Task:** Put bottle cap onto bottle **Skills:** Insertion and alignment, articulated object manipulation **Description:** A bottle without a cap is placed in the workspace. Screw the bottle cap held in hand onto the bottle and tighten it. **Success criteria:** The bottle cap is tightened and remains stationary when a small force is applied in an attempt to rotate the cap. The robot movement stops after terminating the tightening motion, moving away from, and breaking contact with the task objects.
	**Task:** Thread nut onto screw **Skills:** Insertion and alignment, articulated object manipulation **Description:** Screw the nut held in hand onto the screw head placed in the workspace such that their threads fully engage. **Success criteria:** The nut and screw threads engage. The robot movement stops after terminating the rotation motion, moving away from, and breaking contact with the task objects.
	**Task:** Insert key into lock **Skills:** Insertion and alignment **Description:** A lock is placed in the workspace. Insert the key held in hand into the lock. **Success criteria:** The key is inserted into the lock. The robot movement stops after terminating the insertion motion, moving away from, and breaking contact with the task objects.
	**Task:** Plug in charger **Skills:** Insertion and alignment **Description:** An outlet is placed in the workspace. Fully insert the charger held in hand into the outlet. **Success criteria:** The charger is fully inserted into the outlet, such that it does not noticeably move when further pushed toward the outlet. The robot movement stops after terminating the insertion motion, moving away from, and breaking contact with the task objects.

Task Execution Sequence	Task Specs
	**Task:** Place card into holder **Skills:** Placing under environmental constraints **Description:** A card holder that already contains other cards is placed in the workspace. Place the card held in hand into the card holder. **Success criteria:** The robot movement stops after placing the card in the card holder, with existing cards remaining in the card holder. The robot should not remain in contact with the task objects.
	**Task:** Stack cups **Skills:** Placing under environmental constraints **Description:** Multiple cups are scattered around the workspace. Collect the cups and place them inside one another to form a tight stack. **Success criteria:** The robot movement stops after all cups form a tight, upright stack (not knocked over). The robot should not remain in contact with the task objects.
	**Task:** Plug in Ethernet **Skills:** Insertion and alignment **Description:** Fully insert the Ethernet plug held in hand into a socket placed in the workspace. **Success criteria:** The Ethernet plug is fully inserted into the socket with its latch engaged, such that the plug can't be withdrawn by a light pull. The robot movement stops after terminating the insertion motion, moving away from, and breaking contact with the task objects.
	**Task:** Plug in USB **Skills:** Insertion and alignment **Description:** Fully insert the USB stick held in hand into the USB slot placed in the workspace. **Success criteria:** The USB stick is fully inserted into the slot, such that it does not noticeably move when further pushed toward the slot. The robot movement stops after terminating the insertion motion, moving away from, and breaking contact with the task objects.
	**Task:** Turn key to unlock door **Skills:** Articulated object manipulation **Description:** A locked door module is placed in the workspace with a key already inserted into the lock. Turn the key to unlock the door. **Success criteria:** The key is fully turned such that the door can be opened. The robot movement stops after terminating the rotation motion, moving away from, and breaking contact with the task objects.
	**Task:** Turn doorknob to open door **Skills:** Articulated object manipulation **Description:** Turn the doorknob fully to unlock the door, then open the door. **Success criteria:** The door is opened by at least 5 degrees, such that it is visually obvious that the door has been unlocked.
	**Task:** Turn door handle to open door **Skills:** Articulated object manipulation **Description:** Turn the door handle fully to unlock the door, then open the door. **Success criteria:** The door is opened by at least 5 degrees, such that it is visually obvious that the door has been unlocked.
	**Task:** Turn dial to target position **Skills:** Articulated object manipulation **Description:** Operate the dial to turn it to the specified target position. **Success criteria:** The dial is turned to the target position. For dials that can be turned continuously, the final position should be within half a tick interval from the target position. The robot movement stops after terminating the rotation motion, moving away from, and breaking contact with the task objects.

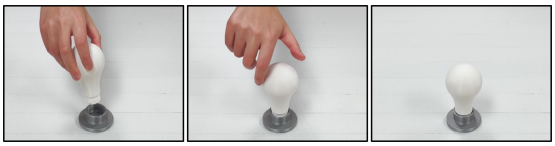
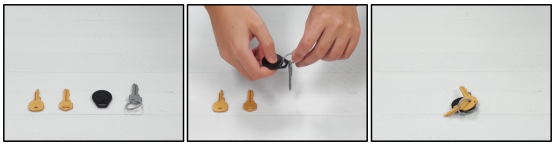
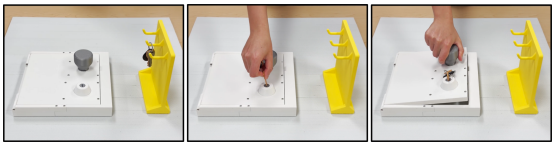
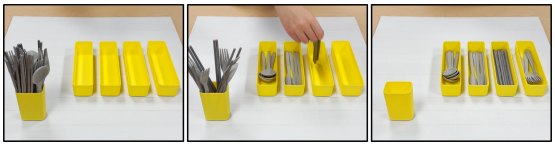
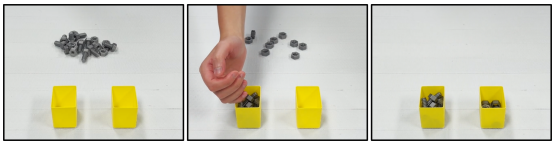
Task Execution Sequence	Task Specs
	**Task:** Install light bulb **Skills:** Insertion and alignment, articulated object manipulation **Description:** A light bulb socket is placed in the workspace. Screw the light bulb held in hand into the socket and tighten it. **Success criteria:** The light bulb is tightened and remains stationary when a small force is applied in an attempt to rotate it. The robot movement stops after terminating the tightening motion, moving away from, and breaking contact with the task objects.

TABLE C2: Longer-horizon benchmark tasks that demand more dexterity and chain several skills together

Task Execution Sequence	Task Specs
	**Task:** Assemble toy block **Skills:** Grasping thin or small objects, grasping from clutter, in-hand repositioning, insertion and alignment, extrinsic contact control **Description:** Components of the block to be assembled, required tools, and bins of screws and nuts are placed in the workspace. Using the tools available, assemble the two halves of the block together with two screws and two nuts. **Success criteria:** The robot movement stops after fully assembling the block with screws tightened and placing the assembled block onto the table. The robot should not remain in contact with the block.
	**Task:** Add keys to keyring **Skills:** Grasping thin or small objects, insertion and alignment **Description:** A keyring and several keys are scattered around the workspace. Add all loose keys onto the keyring. **Success criteria:** The robot movement stops after adding all loose keys to the keyring and placing the chained keys onto the table. The robot should not remain in contact with the task objects.
	**Task:** Unlock and open door **Skills:** Grasping thin or small objects, grasping from clutter, in-hand repositioning, insertion and alignment, articulated object manipulation **Description:** A set of keys on a hanging board and a locked door are placed in the workspace. Retrieve the keys from the hanging board, use the correct key to unlock the door, then turn the doorknob to open the door. **Success criteria:** The door is opened by at least 5 degrees, such that it is visually obvious that the door has been unlocked.
	**Task:** Sort utensils **Skills:** Grasping thin or small objects, grasping multiple objects, grasping from clutter, placing under environmental constraints **Description:** A tight pile of mixed utensils is placed in the workspace or a storage bin in the workspace. Sort the utensils and place different types of utensils into their corresponding storage bins. Being able to hold multiple utensils simultaneously will greatly increase task completion efficiency. **Success criteria:** The robot movement stops after sorting every utensil in the pile into its corresponding bin. Only utensils of the same type should be in the same bin. The robot should not remain in contact with the task objects.
	**Task:** Sort fasteners **Skills:** Grasping thin or small objects, grasping multiple objects, grasping from clutter, placing under environmental constraints **Description:** A tight pile of mixed fasteners is placed in the workspace or a storage bin in the workspace. Sort the fasteners and place different types of fasteners into their corresponding storage bins. Being able to hold multiple fasteners simultaneously will greatly increase task completion efficiency. **Success criteria:** The robot movement stops after sorting every fastener in the pile into its corresponding bin. Only fasteners of the same type should be in the same bin. The robot should not remain in contact with the task objects.