# EMBODIED TASK AGENTS CAN PERFORM FINE MANIPULATION THROUGH TACTILE IN-CONTEXT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Can embodied task agents use tactile experience to perform fine manipulation without task-specific model training? We investigate this capability through Tactile-ETA, a system that incorporates tactile experience into an embodied task agent through in-context learning. Successful expert demonstrations show how contact changes during manipulation. We pair each recorded robot movement with its camera views and bilateral tactile observations. We keep tactile observations at the start and end of each movement and add two neighboring time points when the tactile images change markedly in between. The agent combines these demonstrations with current vision, tactile observations, and proprioception to select actions through general-purpose robot tools, without updating model parameters. Across four simulated manipulation tasks, adding tactile observations to the same two expert demonstrations increases average success from 15.75% to 24.00%, with current sensing and execution interfaces held fixed. Further experiments show that performance depends on the sensory content and representation of demonstrations, while comparing agent models on Lift Bottle reveals substantial differences under the same system configuration. On the real-robot tasks, demonstrations with aligned visual and tactile observations increase the task success rate from 0% to 60%. These findings demonstrate the feasibility of using tactile experience in context for fine manipulation and suggest a path toward extending embodied task agents to additional sensory inputs.

## 1 INTRODUCTION

Embodied task agents such as ETA (Chen et al., 2026d), Show-Harness (Chen et al., 2026c), and RoboDawn (Guo et al., 2026) demonstrate that general-purpose models can execute robot tasks through observations, reasoning, and robot commands. However, fine manipulation remains challenging. Recent real-robot evaluation also shows a substantial performance gap between pick-and-place tasks and fine insertion tasks (Menon et al., 2026). Such manipulation requires adjusting actions to contact conditions, which can be difficult to infer reliably from vision alone. Tactile feedback provides complementary information for these adjustments (Bi et al., 2025). In-context learning further allows demonstrations to provide task guidance at deployment without additional parameter updates (Palo & Johns, 2024b; Yin et al., 2025; Cheng et al., 2026). This creates an opportunity to extend the sensory information these agents can use: *can embodied task agents similarly use tactile experience in context to perform fine manipulation that requires contact-sensitive adjustments?* We investigate this question through **Tactile-ETA** (Figure 1), which incorporates expert tactile experience into an embodied task agent that acts through general-purpose robot tools, without task-specific fine-tuning.

Current tactile observations describe ongoing contact, whereas expert demonstrations can provide a reference for how contact evolves during movement. For example, during rotation or insertion, a demonstration can show how an expert continues moving, adjusts its pose, or changes manipulation stages around a contact transition. Visual–motion records capture what the expert did, but not how contact evolved during the demonstrated actions. We organize expert demonstrations so that the agent can see each movement together with the visual and tactile changes that accompany it.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

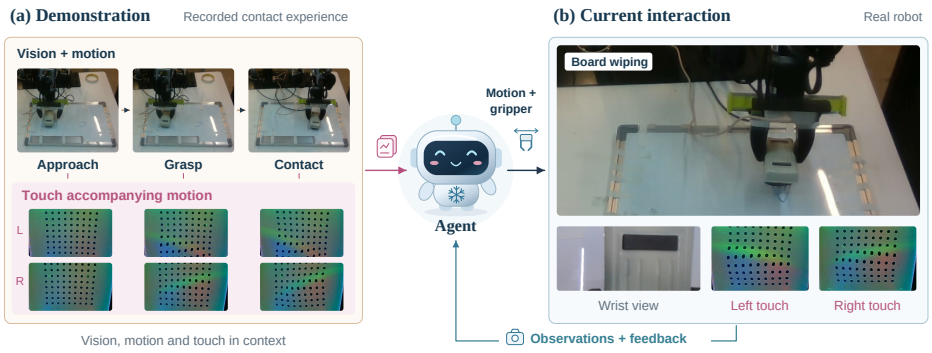



Figure 1: Tactile-ETA performs fine manipulation by conditioning a frozen embodied task agent on (a) action-aligned expert demonstrations that preserve vision, motion, and tactile observations, together with (b) current observations and execution feedback—without parameter updates.

Tactile-ETA divides successful expert demonstrations into action-aligned interaction segments, each containing robot motion, before-and-after camera observations, and bilateral tactile sequences from the same interval.

Full tactile sequences can contain many observations within one movement, while the endpoints alone can miss changes during contact. We therefore retain tactile observations at motion boundaries and, when a change threshold is met, add an adjacent pair spanning the selected change. Complete demonstrations are provided in interaction order as references for execution. The agent combines them with current visual and tactile observations, proprioception, and execution feedback to select motion and gripper commands. Model parameters remain unchanged: task guidance is provided through context, while general-purpose tools and the underlying controller execute the selected actions.

We evaluate the system on four UniVTAC simulation tasks (Chen et al., 2026a), with 100 trials per task and condition. Adding tactile observations to the same two visual–motion demonstrations increases average success from 15.75% to 24.00%, with current sensing and execution interfaces held fixed. Content ablations reveal task-dependent contributions from demonstration vision and tactile observations, while representation ablations show that retaining more tactile frames does not automatically improve performance. A comparison of agent models on Lift Bottle also reveals substantial differences under the same system configuration. Execution statistics and cases illustrate behavioral differences in contact-guided corrections and stage transitions. On two real-robot tasks, a single multimodal demonstration enables three successful trials out of five per task, compared with zero without demonstrations. Together, these results demonstrate the feasibility of using tactile experience in context for embodied task agents and suggest a path toward extending them to additional sensory inputs.

Our contributions are threefold:

- We introduce Tactile-ETA, a system that incorporates expert tactile experience into embodied task agents through context for fine manipulation, without task-specific parameter updates.
- We construct action-aligned visual–motion–tactile demonstrations that compactly retain motion boundaries and intermediate contact changes.
- We show that tactile demonstration experience can improve manipulation performance, with outcomes depending on context organization and the underlying agent model. Execution records illustrate associated differences in contact adjustment and stage transitions.

## 2 Related Work

**Embodied Task Agents.** Language and vision-language models connect task reasoning to robot execution through skill selection, environmental feedback, and program generation (Ahn et al., 2022; Huang et al., 2022; Liang et al., 2023). Visual grounding translates model outputs into spatial targets, objectives, or constraints for robot execution (Huang et al., 2023; 2024b; Liu et al., 2024; Nasiriany et al., 2024; Huang et al., 2024a). Recent work studies general-purpose models operating through robot interfaces (Fu et al., 2026; Chen et al., 2026d;c; Wang et al., 2025; Su et al., 2026). Related systems improve robot programs, skills, or policies through agent-driven iteration (Chen et al., 2026b; Lu et al., 2026; Xiao et al., 2026). We build on OpenETA's tool-based execution (Chen et al., 2026d), in which the agent selects motion and gripper commands and receives observations and feedback after execution.

**In-Context Robot Learning.** Demonstration-guided robot learning (Argall et al., 2009) includes methods that use examples at deployment without further parameter updates (Jiang et al., 2023; Palo & Johns, 2024a; Heppert et al., 2024). Some methods train policies to predict actions conditioned on demonstration trajectories (Fu et al., 2024; Vosylius & Johns, 2025). For frozen general-purpose models, demonstration context can encode visual keypoints and end-effector trajectories as tokens (Palo & Johns, 2024b), or initial object poses and keyframe actions as text (Yin et al., 2025). More closely related to our setting, visual demonstrations and action references are combined with online observations and execution feedback for closed-loop manipulation (Cheng et al., 2026). Interface-aligned demonstrations further illustrate motion-command effects and task strategies (Guo et al., 2026).

**Tactile Sensing for Robotic Manipulation.** Tactile learning supports physical perception and action generation (Guzey et al., 2023). Existing work learns transferable tactile representations (Higuera et al., 2025), connects tactile observations to language-based reasoning (Yu et al., 2024; 2025), and learns sequential visuotactile representations for manipulation (Xu et al., 2026). Policy learning incorporates tactile information through model training (Hao et al., 2025; Huang et al., 2026), with generalist tactile policies extending across sensors (Yuan et al., 2026). Semantic tactile feedback can also support planning alongside a learned controller for action refinement (Bi et al., 2025). Particularly relevant, visual, tactile, and force–torque demonstrations have been used to infer skill sequences and refine their transition conditions, producing a reference plan for subsequent execution and replanning (Chen et al., 2025). Our approach retains expert motions and aligned visual and bilateral tactile observations as demonstration context for successive tool-based decisions. We study how this contact experience can guide an embodied task agent, including controlled comparisons that vary demonstrated tactile information while holding current sensing and execution interfaces fixed.

## 3 Tactile-ETA

### 3.1 Tactile-ETA Overview

Tactile-ETA incorporates tactile experience into an embodied task agent through context, without task-specific parameter updates (Figure 1). The agent receives a task instruction, expert demonstrations, and current visual, tactile, and proprioceptive observations. Expert demonstrations provide the agent with a reference for how the task is successfully executed, including both the expert actions and the contact changes that accompany them, while current observations and execution feedback describe the ongoing interaction. The agent uses both to select calls to general-purpose robot tools; an underlying controller executes the requested movements.

Let $\mathcal{C}_K$ denote the ordered context constructed from $K$ expert episodes, $\ell$ the task instruction, and $\mathcal{I}$ the tool descriptions and static task requirements. Before tool call $t$, the current-episode history $H_t$ contains the observations received so far, previous tool calls, and execution feedback. A model with fixed parameters $\theta$ selects a tool name and arguments $u_t$ according to

$$u_t \sim \pi_\theta(\cdot \mid \ell, \mathcal{I}, \mathcal{C}_K, H_t)\,. \tag{1}$$

The demonstration context is provided once at the start of the episode and remains fixed. After each tool call, returned observations and feedback extend $H_t$ for the next decision, without changing

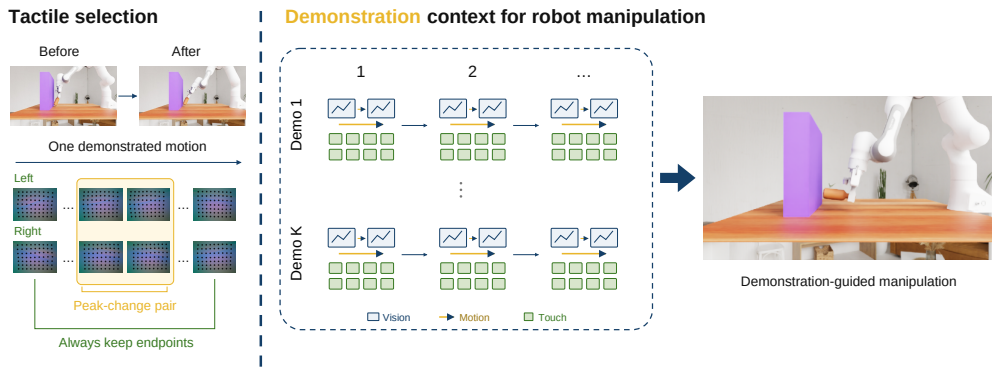



Figure 2: Tactile context construction. (Left) Keep tactile endpoints and, when the threshold is met, the peak-change pair. (Right) Combine tactile observations, vision, and recorded robot movements into ordered demonstrations for robot manipulation.

$\theta$. The following sections describe how we construct the tactile demonstration context and make current tactile feedback available during execution.

### 3.2 Tactile Experience in Context

We construct demonstration context that links expert movements to the visual and tactile observations accompanying them (Figure 2). Each successful expert episode is divided at its recorded action boundaries. Within a segment, before-and-after head- and wrist-camera views describe the scene, while synchronized bilateral tactile observations capture the contact process. The motion record contains measured end-effector translation, relative rotation, and finger-position changes computed from the recorded robot states. These motions and the visual and tactile observations refer to the same action interval. Record fields, units, and coordinate conventions are provided in Appendix B.

**Compact tactile sequences.** We retain tactile observations at both action boundaries. For each fingertip, we select small patches in one tactile image and find their most similar matches at nearby locations in the next image. The distance between each patch's original and matched locations gives its displacement. We sort the displacement magnitudes of reliable matches from smallest to largest and use the value approximately three quarters of the way through this list, the 75th percentile $d_{75}$, to measure the change between the two images. We express $d_{75}$ in original-image pixels and divide it by the selection threshold of 2 pixels to obtain a normalized change score. When too few patches can be matched reliably, we instead use an image-difference score normalized by its own selection threshold (Appendix B).

For a segment with $N \geq 2$ synchronized samples per fingertip, let $c_n$ be the larger normalized score across the two fingertips between samples $n-1$ and $n$. In addition to the endpoints, we retain at most one adjacent pair per segment: the pair with the largest $c_n$, provided that this score is at least one. Let

$$n^\star = \arg\max_{1 \leq n < N} c_n.$$

We retain the time indices

$$\mathcal{S} = \begin{cases} \{0,\ n^\star - 1,\ n^\star,\ N-1\}, & c_{n^\star} \geq 1, \\ \{0,\ N-1\}, & c_{n^\star} < 1. \end{cases} \tag{2}$$

The endpoints associate tactile appearances with the before-and-after visual states; the additional pair preserves an intermediate transition within the same movement. Both fingertips use the same indices, which are sorted and deduplicated, leaving at most four observations per fingertip. A single-sample segment retains its sole observation. The selected images are presented as separate left and right chronological strips with recorded step indices. Selection parameters and an example are given in Appendix B.

**Context assembly.** Each segment pairs these tactile strips with its visual observations and motion record. Segments retain their order within each complete expert episode, together with the recorded episode outcome. The resulting $K$ episodes form $\mathcal{C}_K$, providing a reference for what the expert did and how contact evolved during those movements.

### 3.3 Execution with Current Tactile Feedback

At the beginning of an episode, the embodied task agent receives the demonstration context and selects actions using the task instructions, current visual and tactile observations, and robot state. It interacts through OpenETA's general-purpose robot tools (Chen et al., 2026d). The `move_to` tool specifies an end-effector position and orientation target, together with an optional gripper command; the underlying controller executes the movement. The `observe` tool returns current camera images, bilateral tactile observations, end-effector poses, and joint positions. Demonstrations provide a reference for choosing actions in the current scene.

After each motion segment, the system returns the latest camera images and robot state, along with feedback on whether the target was reached and the deviation from the requested motion. Tactile observations recorded during the movement are selected using the rule in Section 3.2 and presented as separate left and right image strips. The tool call and its returned observations and feedback are appended to the interaction history for subsequent action selection. The agent can therefore refer to both the contact changes in expert demonstrations and the latest feedback from its own actions. Model parameters remain fixed throughout execution. Complete tool schemas and prompts are provided in Appendix F.

## 4 Experiments

We evaluate whether embodied task agents can perform fine manipulation through tactile in-context learning. We examine how demonstrations guide action selection and how performance depends on demonstration content, tactile sequence representation, and the agent model.

**Experimental setup.** We evaluate on four tasks from UniVTAC (Chen et al., 2026a): Insert Tube, Lift Can, Lift Bottle, and Pull Out Key. Corresponding comparisons use the same 100 evaluation initial conditions per task, with success and automatic termination determined by the native environment. Unless otherwise specified, we use a frozen GPT-6 agent with `low` reasoning effort and provide two successful expert episodes from the official dataset when demonstrations are used. Demonstration identities and ordering, robot tools, controller, task instructions, and execution budgets are held fixed within each comparison. Full settings are provided in Appendix A.

### 4.1 Fine Manipulation with Tactile Experience

**Compared configurations.** We compare four input configurations of the same embodied task agent. All receive current vision, proprioception, and execution feedback. *No demos (vision)* provides neither demonstrations nor tactile observations. *No demos (vision + tactile)* adds current tactile observations. *Visual–motion demos* provides current tactile observations and two expert demonstrations containing visual observations and motion records. *Tactile-ETA* further includes the tactile observations from those demonstrations. The latter two configurations use the same expert episodes and current sensory inputs, differing only in whether the demonstrations contain tactile observations.

**Results.** Adding tactile observations to visual–motion demonstrations increases average success across the four tasks from 15.75% to 24.00%, a gain of 8.25 percentage points (Table 1). With current sensory inputs held fixed, this comparison shows that expert tactile experience provides additional guidance for fine manipulation. For reference, average success without demonstrations is 3.00% with current vision alone and 6.25% with current vision and tactile sensing. On Insert Tube, visual–motion demonstrations provide no success-rate gain over the agent with current vision and tactile sensing; adding demonstrated tactile observations improves performance. On Lift Bottle and Pull Out Key, visual–motion demonstrations already improve success, and including demonstrated tactile observations produces a further gain. These results show that visual–motion guidance and

Table 1: Success rates (%) on four UniVTAC tasks, with 100 trials per task and condition. Bold denotes the best result in each row.

| Task | No demos (vision) | No demos (vision + tactile) | Visual–motion demos | Tactile-ETA (ours) |
|---|---|---|---|---|
| Insert Tube | 3 | 15 | 15 | **32** |
| Lift Can | 0 | 1 | **3** | 2 |
| Lift Bottle | 7 | 3 | 27 | **36** |
| Pull Out Key | 2 | 6 | 18 | **26** |
| Average success (%) | 3.00 | 6.25 | 15.75 | **24.00** |

demonstrated tactile experience contribute differently across tasks. Lift Can remains difficult under all configurations.

### 4.2 How Tactile Demonstrations Guide Manipulation

Demonstrations help the agent establish an initial manipulation strategy. On Lift Bottle, the agent with current vision and tactile sensing but no demonstrations begins by closing the gripper in place in only 1/100 episodes, compared with 100/100 under visual–motion guidance. Tactile-ETA adopts the same initial strategy. Expert visual and motion records therefore already provide a concrete starting action for this task.

Tactile experience in demonstrations also influences adjustments after contact. On Insert Tube, 48/100 visual–motion episodes and 13/100 Tactile-ETA episodes end in unsuccessful native early termination with at least two executed segments whose net TCP displacement along the initial approach axis exceeds 1 mm, and no segment with a net displacement below $-1$ mm along that axis (Appendix C.2). Figure 3 shows two examples: the former continues requesting forward motion after observing contact and ultimately fails; the latter reports pronounced tactile distortion, executes a retreat and lateral correction, and subsequently completes insertion. Both agents receive current tactile feedback. Together with the success-rate comparison, these behavioral differences support the use of demonstrated tactile experience to guide action adjustments in response to contact.

### 4.3 What Makes Tactile Context Effective?

We examine demonstration content and tactile sequence representation. The content and representation comparisons use the same two expert episodes and 100 evaluation initial conditions per task, with current vision, tactile observations, proprioception, and execution feedback held fixed. Visual–motion and Tactile-ETA/Selected results are reused from Table 1.

**Demonstration content.** We retain the motion records and compare contexts with no sensory observations, visual observations, tactile observations, or both. Table 2 shows that the contribution of sensory information varies across tasks. On Pull Out Key, tactile–motion demonstrations achieve the same observed success rate as the complete representation. On Lift Bottle, the complete representation performs best, indicating that visual information remains a useful reference for this task.

Table 2: Demonstration content ablation. Success rates (%); bold marks the best per task.

| Context | Lift Bottle | Pull Out Key |
|---|---|---|
| Motion records | 24 | 10 |
| Visual–motion | 27 | 18 |
| Tactile–motion | 26 | **26** |
| Tactile-ETA (ours) | **36** | **26** |

Figure 7 in the appendix illustrates different stage transitions under two demonstration configurations. Both agents follow a rotate–reverse–lift procedure. The agent given motion records alone reverses after its third forward rotation and fails during extraction. The tactile–motion agent performs another rotation after reporting no clear stopping cue, then reverses and pulls after a pronounced tactile change, completing the task. These examples illustrate the distinction between the action

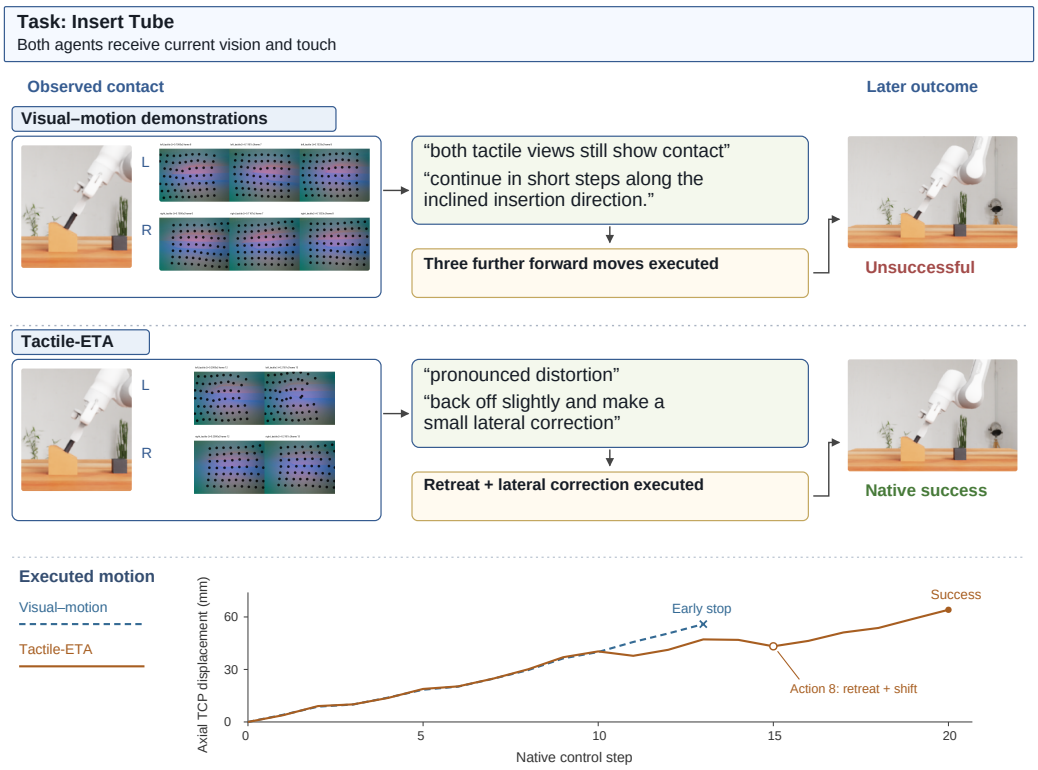



Figure 3: Two Insert Tube rollouts with different demonstration contexts. Visual–motion demonstrations lead to continued advancement and failure; tactile-augmented demonstrations guide retreat and lateral correction, followed by success. Curves show measured TCP displacement along each rollout's initial approach direction.

sequence and the timing of stage transitions, and show how the agent adjusts execution in response to contact changes.

**Tactile sequence representation.** On Lift Bottle, we compare *Endpoints*, retaining only the first and last tactile frames of each demonstrated action; *All frames*, retaining the full recorded sequence; and *Selected*, using the selection rule in Section 3.2. The expert episodes, action segmentation, visual observations, and motion records are unchanged. As shown in Table 3, Selected achieves 36% success, compared with 27% for Endpoints and 2% for All frames. Endpoints matches the success rate of demonstrations without tactile observations. These results show that performance depends on how tactile experience is represented in context.

Table 3: Tactile representation ablation on Lift Bottle. Success rates (%).

| Representation | Success (%) |
|---|---|
| No demo tactile | 27 |
| Endpoints | 27 |
| All frames | 2 |
| Selected (ours) | **36** |

Figure 4 illustrates the intermediate tactile information retained by Selected in an Insert Tube demonstration. Despite similar endpoint patterns, the selected intermediate frame shows a wider blue band, revealing a change that endpoint-only sampling omits.

With one, two, and four complete demonstrations, mean success rates are 22.50%, 24.00%, and 19.00%, respectively (Appendix D). Increasing from two to four raises Pull Out Key success from 26% to 48% but reduces Lift Bottle success from 36% to 2%.

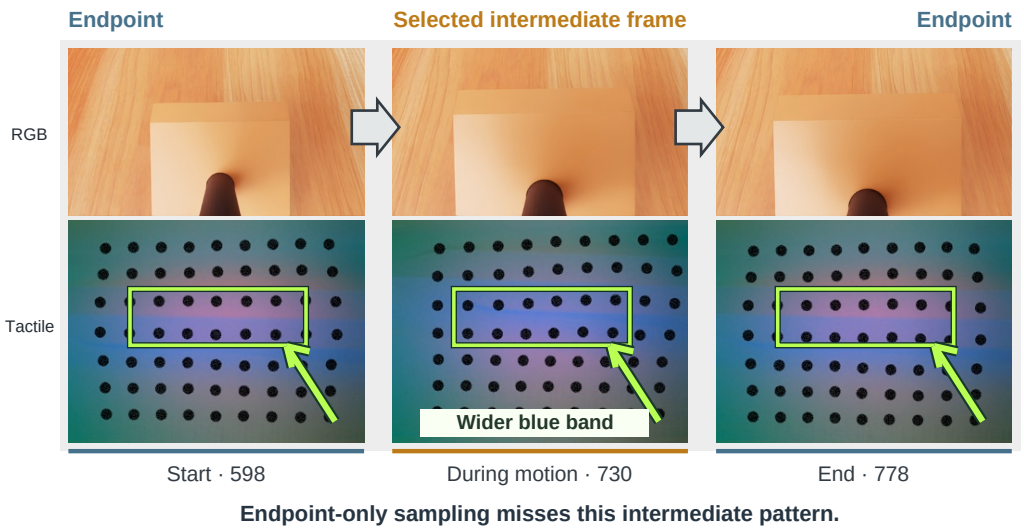



Figure 4: **Intermediate tactile information missed by endpoint-only sampling.** Columns show the start, a selected intermediate frame, and the end of an Insert Tube demonstration segment, with synchronized RGB observations above and right-fingertip tactile images below. Although the endpoint tactile patterns look similar, the intermediate frame exhibits a wider blue band in the highlighted region. Endpoint-only sampling omits this intermediate pattern, whereas Selected retains it in the demonstration context. Green boxes and arrows identify the same image region for comparison. One of the two selected intermediate frames is shown.

### 4.4 Dependence on the Agent Model

We compare four agent models on Lift Bottle using the same two tactile-augmented demonstrations and 100 evaluation initial conditions. All models use the `low` reasoning-effort setting, with the demonstration representation, task instructions, observation modalities, robot tools, controller, and execution budgets held fixed.

Figure 5 shows substantial differences in manipulation performance despite identical demonstrated experience and execution interfaces. These results indicate that tactile context design should be considered together with the agent's ability to use observations, demonstrations, and feedback for decision-making.

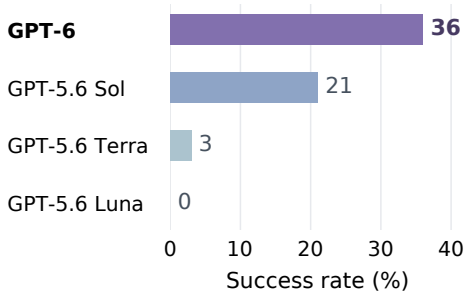



Figure 5: Agent model comparison on Lift Bottle: 100 trials per model with the same two demonstrations. GPT-6 results are reused from Table 1.

### 4.5 Real-World Experiments

We further evaluate Tactile-ETA on a real robot with two tasks: *Wipe Board* and *Pick Up Chips*. The platform uses a reBot Arm B601-RS equipped with a vitai GF225 tactile gripper. Visual observations come from an Orbbec wrist-mounted camera and a RealSense D435i third-person camera. The former requires wiping a whiteboard to remove the target writing; the latter requires lifting a potato chip without breaking or dropping it. We compare execution without demonstrations against execution with one visual–motion–tactile demonstration. Both configurations receive current vision, tactile observations, and robot state. The demonstration is provided through context, with model parameters held fixed. Figure 6(a) illustrates the two tasks. Videos of the real-world experiments are provided in the supplementary material. Detailed settings and success criteria are provided in Appendix E.

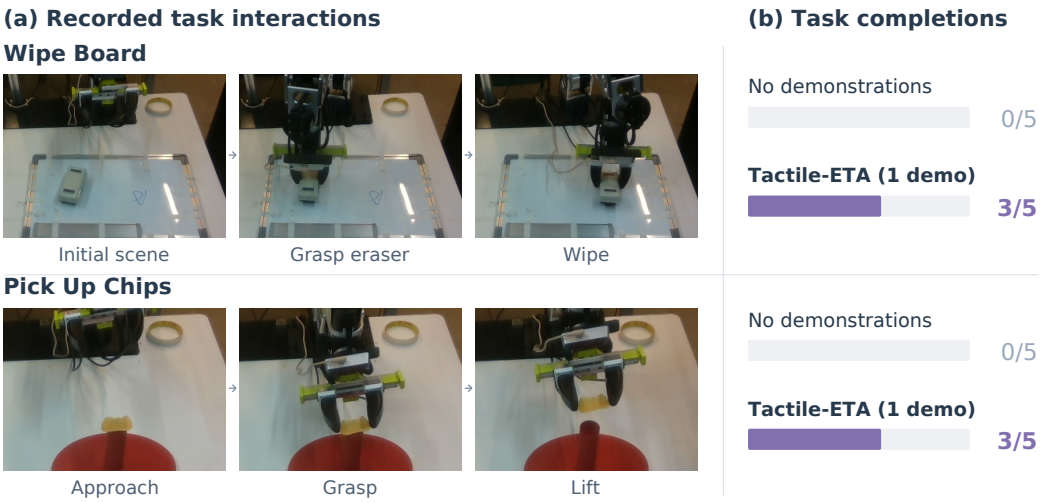



Figure 6: Real-world tasks and results. (a) Recorded interactions illustrate wiping and chip pickup. (b) Task completions over five trials per configuration. Both receive current vision and tactile observations; Tactile-ETA additionally receives one multimodal human demonstration.

As shown in Figure 6(b), the agent without demonstrations completes neither task in any of the five trials, while Tactile-ETA completes three of five trials on each task. These results extend our findings on demonstration experience to real robots: with current tactile sensing available, a single multimodal demonstration provides useful guidance for manipulation. Together with the controlled comparisons of demonstrated tactile information in simulation, the real-world results further demonstrate the feasibility of incorporating tactile experience into embodied task agents through context.

## 5 Discussion

**Limitations and future directions.** Tactile-ETA explores how incorporating touch as an additional input modality through in-context experience can extend embodied task agents' capabilities for fine manipulation. We hope this work encourages broader exploration of how new sensory modalities can expand agents' manipulation capabilities. Our evaluation is limited in model coverage by API availability and inference cost; broader community evaluation could help characterize how different foundation models use tactile experience. In addition, the current system primarily uses tactile feedback to guide decisions between motion segments. Combining decisions informed by tactile experience with fast tactile feedback control during action execution is a promising direction for improving continuous contact-rich interaction.

**Conclusions.** We introduced Tactile-ETA, which enables embodied task agents to use tactile experience in context for fine manipulation without updating model parameters. The system organizes expert demonstrations into compact, action-aligned visual, motion, and tactile context and combines this experience with current sensing and execution feedback. Across four simulated tasks, adding tactile observations to visual–motion demonstrations improves average success while retaining the same current sensory inputs. Further experiments show that performance depends on demonstration content, tactile representation, and the agent model. Real-robot experiments demonstrate successful wiping and fragile-object pickup with a single multimodal demonstration. These findings establish the feasibility of tactile in-context learning for fine manipulation and suggest a path toward extending the sensory capabilities of embodied task agents through experience provided in context.

### AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and revision, code editing, and figure layout. Experimental photographs in the figures come from recorded simulation or robot observations.

### REPRODUCIBILITY STATEMENT

Appendix A specifies the simulation inputs, execution budgets, and native evaluation criteria. Appendix B shows a complete expert action segment and documents tactile selection and input packaging. Appendix C defines the behavioral counts and identifies the analyzed episodes; Appendix D reports the demonstration-count results and trial accounting. Appendix E describes the real-robot evaluation, and Appendix F provides the retained simulation prompts and tool schemas. The corresponding segment record, behavioral definitions, and demonstration-count summaries are included in `appendix_materials/`.

## REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL https://arxiv.org/abs/2204.01691.

Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2008.10.024. URL https://www.sciencedirect.com/science/article/pii/S0921889008001772.

Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. Vla-touch: Enhancing vision-language-action models with dual-level tactile feedback, 2025. URL https://arxiv.org/abs/2507.17294.

Baijun Chen, Weijie Wan, Tianxing Chen, Xianda Guo, Congsheng Xu, Yuanyang Qi, Haojie Zhang, Longyan Wu, Tianling Xu, Zixuan Li, Yizhe Wu, Rui Li, Xiaokang Yang, Ping Luo, Wei Sui, and Yao Mu. Univtac: A unified simulation platform for visuo-tactile manipulation data generation, learning, and benchmarking, 2026a. URL https://arxiv.org/abs/2602.10093.

Kaiyuan Chen, Shuangyu Xie, Letian Fu, Justin Yu, William Pacini, Sandeep Bajamahal, Hudson Kim, Jaimyn Drake, Daehwa Kim, Haoru Xue, Jonathan Francis, Christian Juette, Peter Schaldenbrand, Muhammet Yunus Seker, Ruwan Wickramarachchi, Uksang Yoo, Guanzhi Wang, Adithyavairavan Murali, Balakumar Sundaralingam, S. Shankar Sastry, Spencer Huang, Yuke Zhu, Linxi "Jim" Fan, and Ken Goldberg. Gap: A graph-as-policy multi-agent self-learning harness for variational automation tasks, 2026b. URL https://arxiv.org/abs/2607.05369.

Kejia Chen, Zheng Shen, Yue Zhang, Lingyun Chen, Fan Wu, Zhenshan Bing, Sami Haddadin, and Alois Knoll. Lemmo-plan: Llm-enhanced learning from multi-modal demonstration for planning sequential contact-rich manipulation tasks, 2025. URL https://arxiv.org/abs/2409.11863.

Yanzhe Chen, Zechen Bai, Zhijun Cao, Wenzheng Zeng, Kevin Qinghong Lin, Yiqi Lin, Guoqiang Liang, Kevin Yuchen Ma, Qiming Huang, and Mike Zheng Shou. Show-harness: Just a vlm agent can play robots, 2026c. URL https://arxiv.org/abs/2609.10522.

Yitong Chen, Zezheng Huai, Sixian Li, Yubang Wang, Haozhe Zhang, Yifei Zhang, Hechang Chen, Jingjing Gong, Yu-Gang Jiang, and Xipeng Qiu. Eta: A new agentic paradigm for embodied tasks, 2026d. URL https://arxiv.org/abs/2608.03924.

Dongzhou Cheng, Taoran Yi, Ye Fang, Xingwu Zhang, Fan Feng, Yixuan Li, Gengxiong Zhuang, Rongze Wang, Shuai Yang, Wei Song, Weizhi Xue, Minyan Wu, Jie Gui, Jiaqi Wang, and Tong Wu. In-context robot learning with vlm agents, 2026. URL https://arxiv.org/abs/2609.19138.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction, 2024. URL https://arxiv.org/abs/2408.15980.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Dantong Niu, Fei-Fei Li, Guanya Shi, Jiajun Wu, Shankar Sastry, Yuke Zhu, Ken Goldberg, and Linxi "Jim" Fan. Cap-x: A framework for benchmarking and improving coding agents for robot manipulation, 2026. URL https://arxiv.org/abs/2603.22435.

Meng-Hao Guo, Zhe-Han Mo, Jia-Jun Wang, Yi Zhang, Kejin Wang, Yi-Xuan Deng, Jia-Peng Zhang, Yongming Rao, and Shi-Min Hu. Transferring the intelligence of vlms to robotic control, 2026. URL https://arxiv.org/abs/2609.22966.

Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. See to touch: Learning tactile dexterity through visual incentives, 2023. URL https://arxiv.org/abs/2309.12300.

Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Tla: Tactile-language-action model for contact-rich manipulation, 2025. URL https://arxiv.org/abs/2503.08548.

Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Ditto: Demonstration imitation by trajectory transformation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7565–7572. IEEE, October 2024. doi: 10.1109/iros58592.2024.10801982. URL http://dx.doi.org/10.1109/IROS58592.2024.10801982.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 885–915. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/higuera25a.html.

Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models, 2024a. URL https://arxiv.org/abs/2403.08248.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models, 2022. URL https://arxiv.org/abs/2207.05608.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models, 2023. URL https://arxiv.org/abs/2307.05973.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation, 2024b. URL https://arxiv.org/abs/2409.01652.

Yuzhe Huang, Pei Lin, Wanlin Li, Daohan Li, Jiajun Li, Jiaming Jiang, Chenxi Xiao, and Ziyuan Jiao. Taf-vla: Tactile-force alignment in vision-language-action models for force-aware manipulation, 2026. URL https://arxiv.org/abs/2601.20321.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023. URL https://arxiv.org/abs/2210.03094.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. URL https://arxiv.org/abs/2209.07753.

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting, 2024. URL https://arxiv.org/abs/2403.03174.

Runyu Lu, Yubo Wu, Ethan Kou, Letian Fu, Wenli Xiao, Ajay Mandlekar, Yinzhen Xu, Guanya Shi, Ken Goldberg, Ang Chen, Mosharaf Chowdhury, Yuke Zhu, Linxi "Jim" Fan, and Guanzhi Wang. Aspire: Agentic /skills discovery for robotics, 2026. URL https://arxiv.org/abs/2607.00272.

Achu Menon, Sravanthi Machcha, Sabrina Zou, Tzu Kit Chan, and Jay Chooi. Gpt-6 astra on robotic manipulation. https://openai.robocurve.org/gpt-6-astra/, September 2026. Robocurve, accessed September 26, 2026.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. Pivot: Iterative visual prompting elicits actionable knowledge for vlms, 2024. URL https://arxiv.org/abs/2402.07872.

Norman Di Palo and Edward Johns. Dinobot: Robot manipulation via retrieval and alignment with vision foundation models, 2024a. URL https://arxiv.org/abs/2402.13181.

Norman Di Palo and Edward Johns. Keypoint action tokens enable in-context imitation learning in robotics, 2024b. URL https://arxiv.org/abs/2403.19578.

Jiayi Su, Yixin Zheng, Mi Yan, Li Yi, Zhizheng Zhang, and He Wang. GPT 6 Astra as an embodied policy. Technical report and code, 2026. URL https://github.com/anonymous-report-421/eval-of-gpt-6-astra-as-policy.

Vitalis Vosylius and Edward Johns. Instant policy: In-context imitation learning via graph diffusion, 2025. URL https://arxiv.org/abs/2411.12633.

Zidan Wang, Rui Shen, and Bradly Stadie. Wonderful team: Zero-shot physical task planning with visual llms, 2025. URL https://arxiv.org/abs/2407.19094.

Wenli Xiao, Jia Xie, Tonghe Zhang, Haotian Lin, Letian "Max" Fu, Haoru Xue, Jalen Lu, Yi Yang, Cunxi Dai, Zi Wang, Jimmy Wu, Guanzhi Wang, S. Shankar Sastry, Ken Goldberg, Linxi "Jim" Fan, Yuke Zhu, and Guanya Shi. Enpire: Agentic robot policy self-improvement in the real world, 2026. URL https://arxiv.org/abs/2606.19980.

Congsheng Xu, Qiaochu Yang, Fangyuan Shi, Yifan Han, Baijun Chen, Yiming Wang, Haonan Zhao, Daolin Ma, Xiaokang Yang, and Hesheng Wang. Vt-muse: Multimodal unified sequential visuotactile representation learning for manipulation, 2026. URL https://arxiv.org/abs/2608.21290.

Yida Yin, Zekai Wang, Yuvan Sharma, Dantong Niu, Trevor Darrell, and Roei Herzig. In-context learning enables robot action prediction in llms, 2025. URL https://arxiv.org/abs/2410.12782.

Samson Yu, Kelvin Lin, Anxing Xiao, Jiafei Duan, and Harold Soh. Octopi: Object property reasoning with large tactile-language models, 2024. URL https://arxiv.org/abs/2405.02794.

Samson Yu, Kelvin Lin, and Harold Soh. Demonstrating the octopi-1.5 visual-tactile-language model, 2025. URL https://arxiv.org/abs/2507.09985.

Chengbo Yuan, Zicheng Zhang, Mingjie Zhou, Wendi Chen, Yi Wang, Zhuoyang Liu, Dantong Niu, Shuo Wang, Hui Zhang, Wenkang Zhang, Yingdong Hu, Yuanqing Gong, Wanli Xing, Chuan Wen, Cewu Lu, Kaifeng Zhang, and Yang Gao. Ftp-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation, 2026. URL https://arxiv.org/abs/2606.13102.

# A EXPERIMENTAL SETUP

We detail the simulation protocol and controlled inputs here. Appendix B describes demonstration construction; Appendix F provides the operator prompts and tool schemas.

## A.1 SIMULATION SETUP

We evaluate the main input configurations on four UniVTAC tasks: Insert Tube, Lift Can, Lift Bottle, and Pull Out Key. Each configuration is evaluated on the same 100 initial conditions per task, using separate rollouts. The agent begins operating after the environment's native reset and preparatory motion. From this point onward, it selects the motion and gripper commands through the robot tools. Success rates are computed using the native task criteria and reported both per task and averaged across tasks.

The main comparisons use a frozen GPT-6 agent with the `low` reasoning-effort setting. Demonstration-based conditions receive the same two successful official expert episodes, 0 and 1, in that order. One demonstration denotes a complete expert episode rather than a single action segment. The demonstration-content ablation covers Lift Bottle and Pull Out Key. The tactile-representation comparison in Table 3 concerns Lift Bottle, and the model comparison is reported in Section 4.4.

## A.2 INPUT CONFIGURATIONS

Table 4 summarizes the information available in the main comparisons and demonstration-content ablations. All configurations retain current camera observations, proprioception, and ordinary execution feedback. Removing images from the demonstrations does not remove the corresponding current observations. Only the no-demonstration vision baseline omits current tactile observations.

Table 4: Input configurations. Numerical demonstration records contain recorded robot movements, robot states, timing, and episode outcomes. All settings retain current camera observations, proprioception, and ordinary execution feedback. Motion records and Tactile–motion are additional content ablations evaluated on Lift Bottle and Pull Out Key.

| Setting | Current tactile | Demonstration records | Demonstration vision | Demonstration tactile |
|---|---|---|---|---|
| No demos (vision) | – | – | – | – |
| No demos (vision + tactile) | Yes | – | – | – |
| Motion records | Yes | Yes | – | – |
| Visual–motion | Yes | Yes | Yes | – |
| Tactile–motion | Yes | Yes | – | Yes |
| Tactile-ETA | Yes | Yes | Yes | Yes |

For the four demonstration-content settings, the expert episodes, segment boundaries, recorded robot movements, robot states, timing, and outcomes remain the same. Only the images in the demonstrations are varied. The tactile–motion condition retains the same bilateral tactile sequences as the complete representation, while omitting the camera images and their references from the demonstrations.

## A.3 EXECUTION SETTINGS

The agent issues motion targets through `move_to` and obtains observations through `observe`. Camera views, robot state, enabled tactile history, and target-tracking feedback are returned after execution. Table 5 lists the controller and episode budgets; exact interface conventions are given in Appendix F.

Table 5: Execution settings in the retained prompt and configuration records.

| Setting | Value |
|---|---|
| Physics / control frequency | 120 / 60 Hz (decimation 2) |
| Maximum control steps per motion | 80 |
| Admitted non-preview motions | 30 per episode |
| MCP calls | 100 per episode |
| Total operator wall time | 3600 s |
| Post-terminal observation/completion grace | Up to 300 s within total budget |
| Insert Tube / Lift Can control-step limit | 300 / 300 |
| Lift Bottle / Pull Out Key control-step limit | 500 / 300 |
| Position / orientation arrival tolerance | 0.003 m / 0.05236 rad |
| Maximum joint change per control step | 0.025 rad |
| Gripper settle time / tolerance | 0.1 s / 0.0002 m |
| Gripper target tolerance / maximum wait | 0.001 m / 40 control steps |

### A.4 Native evaluation and termination

Native success checks, early-failure conditions, and episode budgets determine when physical interaction ends. These checks run in the background. The agent is not given current task scores or the specific native termination reason, and no `check_task` tool is available. Automatic termination is reported with the neutral notification `episode_ended`. The `finish_episode` tool is used to end voluntarily or acknowledge termination. Historical expert outcomes and static task requirements remain available; they are distinct from evaluation of the current rollout. Success rates use the environment's recorded labels, not the agent's final account of its performance.

We retain the benchmark's task definitions. In particular, Lift Can is evaluated using its native pose predicate, not a separate criterion for stable suspended lifting.

## B Demonstration Construction

Each recorded action segment pairs robot movement and boundary robot states with before-and-after camera views and selected bilateral tactile observations, following Section 3.2. Complete expert episodes retain segment order and recorded outcomes. The same two episodes are supplied once at rollout start. We first show a complete recorded action segment, then specify tactile selection and the formats used to present the resulting observations.

### B.1 A complete action-segment example

We show segment 5 of Lift Bottle expert episode 0, spanning recorded steps 392–464. The complete segment record, including both boundary poses and joint states, the tactile selection record, and four associated image files, is supplied in `appendix_materials/segment_example.json`. This is one segment of a demonstration, not a complete episode or the entire two-demonstration tool response. Numerical values below are rounded for readability; the JSON retains their original precision.

**Lift Bottle: episode 0, segment 5**

**Boundary states.** Steps 392 $\rightarrow$ 464; hand position (m): $[0.55427, -0.04652, 0.21738] \rightarrow [0.57888, -0.04511, 0.24504]$.

**Measured movement.** Translation: $[24.61, 1.41, 27.65]$ mm; rotation vector: $[-4.89, 2.40, 2.88] \times 10^{-6}$ rad; finger changes: $[1.68, -1.68] \times 10^{-8}$ m; nominal duration: 0.60 s.

**Tactile times.** $[392, 428, 430, 464]$, shared by both fingertips.

**Feedback.** "Observed state transition; original execution feedback is not separately recorded."

**Lift Bottle: episode 0, segment 5 (continued)**

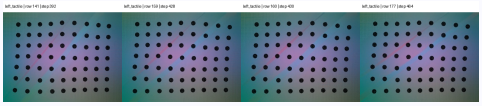
`example 0 segment 5`
`head before/after`



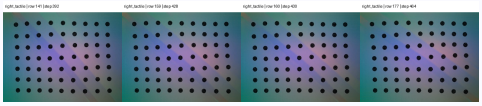
`example 0 segment 5`
`wrist before/after`



`example 0 segment 5`
`left_tactile history`



`example 0 segment 5`
`right_tactile history`



Historical poses use robot-base-relative `panda_hand` positions in metres and quaternions in `wxyz` order. Joint states contain seven arm angles in radians and two finger positions in metres. Nominal duration is the recorded step difference divided by 120, rather than a measured wall-clock interval. These measured movements are not recorded online TCP commands.

The headings reproduce the image labels referenced by the segment's `vision_labels` and its associated `historical_touch` record. The tool interface supplies these labels in text alongside the image blocks; local image paths are packaging locators and are omitted from the transmitted descriptors. The strips above are the prepared input images, including their recorded-step labels. They preserve the two camera boundaries and the common four-time tactile sequence without adding force or slip annotations.

### B.2 Tactile selection parameters

Images are converted to grayscale and resized while preserving their aspect ratio. We track a $3 \times 4$ grid of patches inside the region of interest (ROI). Each patch is matched by an integer displacement search minimizing mean squared error (MSE). Low-texture patches and ambiguous or high-error matches are rejected. Tracking quality is the fraction of the 12 patches that survive. The motion score is the 75th percentile of their displacement magnitudes, rescaled to the original image resolution. When tracking quality is below 0.35, we instead use the mean absolute grayscale difference inside the ROI, divided by 255. Table 6 gives the retained selection parameters for the expert demonstrations and online tactile history.

Table 6: Tactile selection parameters. Spatial tracking settings use resized pixels; the motion threshold uses original-image pixels.

| Parameter | Value |
|---|---|
| Tracking width | 160 pixels |
| ROI $(x_0, y_0, x_1, y_1)$ | $(0.08, 0.08, 0.92, 0.92)$ of image size |
| Tracking grid | 3 rows $\times$ 4 columns |
| Patch radius / patch size | 6 / 13 $\times$ 13 pixels |
| Search radius | 4 pixels ($9 \times 9$ candidates) |
| Minimum patch standard deviation | 8.0 grayscale levels |
| Maximum match MSE | 900.0 |
| Minimum second-best minus best MSE | 1.0 |
| Minimum tracking quality | 0.35 |
| Motion-change threshold | 2.0 original-image pixels |
| Image-difference threshold | 0.015 (normalized intensity) |

Each fingertip's score is divided by the threshold for its active measure. The larger bilateral score determines the peak transition. Ties select the first maximum in chronological order. Endpoints are always retained; the two samples immediately before and after the peak transition are added only

when its normalized score reaches one. Indices are shared across fingers, sorted and deduplicated. A single-sample segment retains its sole sample.

### B.3 Tactile representation and input packaging

Figure 4 shows a qualitative example from an Insert Tube expert demonstration. Selected retains the segment endpoints at recorded steps 598 and 778 and the adjacent intermediate observations at steps 728 and 730. The figure displays the endpoints and step 730, with synchronized wrist-camera and right-fingertip observations. Green annotations mark the same tactile image region in each column. The wider blue band in the intermediate frame illustrates information omitted by endpoint-only sampling.

## C Behavioral Evidence and Case Studies

We present selected episodes to connect demonstration content with observable agent decisions. Each example relates the returned observations, the agent's public messages, its subsequent tool calls, and the resulting execution. Quotations report the agent's interpretation; executed movements and native task outcomes are recorded separately.

**Behavioral counting.** The behavioral counts in Section 4 are computed by scripts from the locked episode records, rather than from the agent's self-reported intentions. Each denominator includes all 100 query episodes for the stated task and condition, irrespective of success. Execution events are host tool-trace entries containing a resolved requested target and execution feedback; previews are excluded.

Table 7 summarizes the analyzed cases.

Table 7: Cases included in the qualitative analyses.

| Case | Task | Setting |
|---|---|---|
| Target arrival without success | Lift Bottle | Tactile–motion |
| Uprighting (success) | Lift Bottle | Selected |
| Uprighting (failure) | Lift Bottle | Endpoints |
| Supplementary success | Lift Bottle | Endpoints |
| Supplementary failure | Lift Bottle | Selected |
| Contact adjustment | Insert Tube | Visual–motion |
| Contact adjustment | Insert Tube | Tactile-ETA |
| Stage transition (success) | Pull Out Key | Tactile–motion |
| Stage transition (failure) | Pull Out Key | Motion records |

### C.1 Initial grasp and object outcomes on Lift Bottle

We examine the first execution event. It qualifies when its requested gripper command is `close`, the norm of its resolved target translation relative to the pre-execution TCP is at most 1 mm, and its requested orientation change is at most $1^\circ$. An episode with no execution event does not qualify. This measures the initial requested strategy, not whether the fingers actually closed or a stable grasp was achieved. The counts are 1/100 without demonstrations (with current tactile input), 100/100 with visual–motion demonstrations, and 100/100 with Tactile-ETA.

The tactile–motion failure case illustrates the difference between reaching a robot target and completing object manipulation. All 24 executed motion requests report target arrival, yet the task remains unsuccessful. The recorded views show the bottle still inclined after partial reorientation. Opening and lowering the gripper, followed by another closure, leaves the bottle back on the surface; subsequent regrasp attempts do not complete the task.

This episode distinguishes controller target arrival from successful object manipulation: reaching the requested end-effector poses does not ensure that the resulting object pose and grasp support the next stage.

**Uprighting and recovery in executed episodes.** The successful Selected episode and the failed Endpoints episode both begin with grasping, lifting, and an initial orientation adjustment. Their next movements differ. The Selected agent maintains its wrist orientation and continues with translations that bring the bottle upright. The Endpoints agent requests additional rotation together with translation. This movement misses its position target, the bottle falls back onto the surface, and subsequent recovery remains unsuccessful.

A successful initial grasp does not determine the outcome of later reorientation. These episodes have different initial conditions and illustrate execution patterns rather than a matched causal comparison.

An additional successful Endpoints episode also continues with translations after an orientation adjustment. A Selected failure case instead loses the object during further reorientation. These counterexamples show that the representation does not prescribe a universally successful action sequence.

### C.2 Contact-guided adjustment on Insert Tube

For each executed segment, we project the measured net TCP displacement onto that episode's initial `approach_world` axis. We count an episode if it fails with `native_early_stop`, contains at least two segments with projected displacement greater than 1 mm, and contains no segment below $-1$ mm. The forward segments need not be adjacent; all executed segments are considered. Displacements exactly at the thresholds do not qualify as forward or retreating. This is a segment-level measure, not a claim that no backward motion occurred within any segment. The counts are 48/100 with visual–motion demonstrations and 13/100 with Tactile-ETA. The definitions, thresholds, and counts are also supplied in `appendix_materials/behavioral_counts.json`.

Figure 3 contrasts visual–motion and tactile-augmented demonstrations for the same initial condition. Both conditions retain current vision, tactile observations, and proprioception. In the visual–motion example, the agent continues advancing until unsuccessful early termination. In the tactile-augmented example, it adjusts its pose and subsequently retreats and shifts laterally before completing the task.

After receiving the tactile observations following action 7, the tactile-augmented agent states:

> **Tactile-ETA** Insert Tube / after action 7
>
> "The left tactile image now shows pronounced distortion. I'll back off slightly and make a small lateral correction before advancing again."

The next motion request specifies a world-frame displacement of $[-2, -2, 3.46]$ mm. Its recorded execution includes the retreat and lateral correction illustrated in Figure 3. The agent subsequently resumes advancing and reaches native success. This sequence shows the link between the stated contact observation and the corrective movement actually executed.

### C.3 Stage transitions on Pull Out Key

We examine a successful episode with tactile–motion demonstrations and a failed episode with numerical motion records alone. Both agents receive current visual and tactile observations and proprioception. Their expert demonstrations use the same two expert episodes and follow a rotate–reverse–lift procedure. The difference lies in when the agents decide to transition from rotation to pulling.

After its third forward rotation, the agent with tactile–motion demonstrations states:

> **Tactile–motion** Pull Out Key / after the third forward rotation
>
> "The key remains held between both tactile pads, and the rotation has progressed without a clear stop yet. I'll continue in smaller increments while keeping the TCP centered over the slot."

It then requests and executes a fourth forward rotation. After receiving the next observation, it states:

> **Tactile–motion** Pull Out Key / after the fourth forward rotation
>
> "The latest turn produced a pronounced change in the left contact patch, consistent with loading against the rotational stop. I'll ease the rotation back slightly, then pull vertically while preserving the grasp."

Its subsequent tool calls reverse the rotation and request a 30 mm upward motion. The episode ends in native success.

The agent given numerical motion records alone also describes the current tactile images. After its third forward rotation, it states:

> **Motion records** Pull Out Key / after the third forward rotation
>
> "Both tactile images show pronounced deformation during the turn, consistent with loading against resistance. I'll ease the rotation back slightly, then pull vertically while keeping the grasp."

It then reverses the rotation and requests a 30 mm upward motion. This episode ends in failure.

The cases differ in when rotation gives way to pulling, despite both agents receiving current tactile observations. They illustrate stage-transition decisions; the aggregate comparison is reported in Table 2.

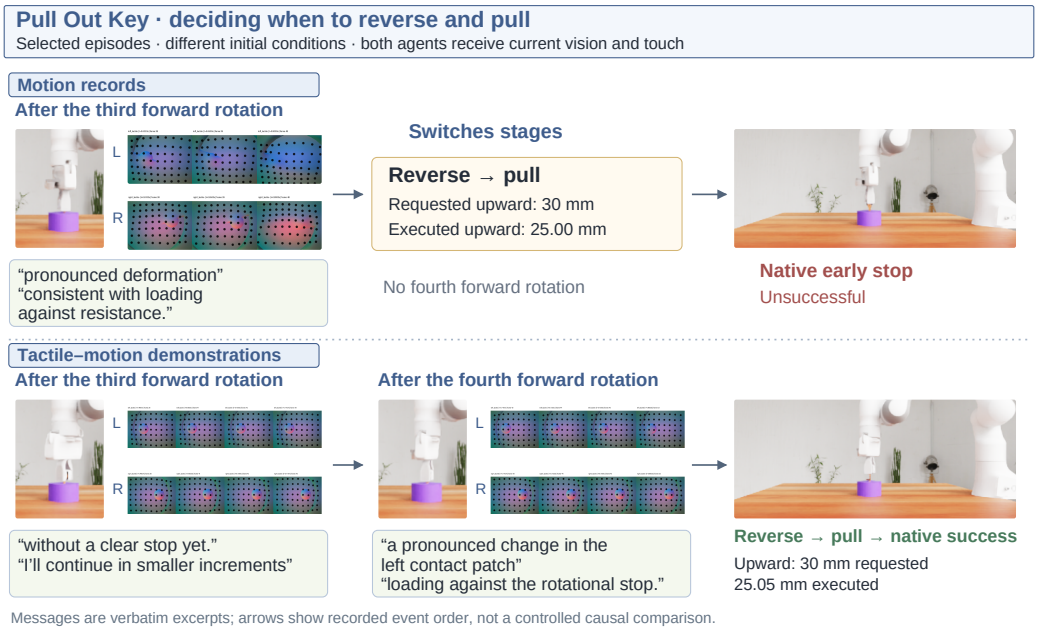



Figure 7: Stage transitions in two Pull Out Key episodes. The motion-record agent reverses after three forward rotations; the tactile–motion agent performs a fourth before reversing and pulling. Images and message excerpts show the recorded sequence. Initial conditions differ; outcomes are native labels.

## D Effect of Demonstration Count

We vary the number of complete expert episodes supplied to Tactile-ETA, using GPT-6 Astra with `low` reasoning effort and the same four tasks and 100 initial conditions per task as the main comparison. Each demonstration includes visual observations, bilateral tactile observations, and numerical records. The 2-shot condition uses expert episodes 0 and 1; 4-shot adds the next two successful official episodes in ID order. For 1-shot, even-indexed queries use episode 0 and odd-indexed queries use episode 1, with 50 queries each. Current sensing and the execution interface are unchanged.

Table 8: Success rate (%) with 1, 2, or 4 complete demonstrations. Each task–setting pair contains 100 evaluable query episodes. The shaded column denotes the 2-shot setting used in the main comparisons.

| Task | 1-shot | 2-shot | 4-shot |
|---|---|---|---|
| Insert Tube | 22 | **32** | 23 |
| Lift Can | 2 | 2 | **3** |
| Lift Bottle | 31 | **36** | 2 |
| Pull Out Key | 35 | 26 | **48** |
| Average | 22.50 | **24.00** | 19.00 |

More demonstrations do not consistently improve performance (Table 8). Increasing from two to four demonstrations improves Pull Out Key but reduces success on Insert Tube and Lift Bottle.

**Trial accounting.** The first pass produced 1,086 evaluable outcomes. A separate supplement completed 114 missing query–setting combinations after initialization unavailability (104), infrastructure issues (6), or cancellation (4). The supplement removed initialization and worker-startup deadlines while retaining operation and cleanup budgets. Existing valid task failures were not rerun. All rates use the resulting 100 evaluable outcomes per task and shot count, combining the two initialization policies. Supplement counts for 1/2/4-shot are 7/6/6 on Insert Tube, 26/31/18 on Lift Can, 5/3/1 on Lift Bottle, and 2/4/5 on Pull Out Key. Machine-readable counts and the selection protocol are provided in `appendix_materials/shot_scaling_results.json`.

# E Real-World Experiments

**Robot platform.** Both tasks use a reBot Arm B601-RS with a vitai GF225 tactile gripper.

**Agent and demonstrations.** Both tasks use GPT-6 Astra with `low` reasoning effort. The no-demonstration and Tactile-ETA configurations both receive current visual and tactile observations and robot state. Tactile-ETA additionally receives one manually recorded demonstration per task containing visual observations, robot motion, and tactile observations. The demonstration is supplied in context without updating model parameters. Each task is evaluated with five trials per configuration.

**Success criteria.** Task success is judged by a human evaluator. For Wipe Board, success requires removing the target blue writing from the whiteboard, with the eraser moved away so that the cleaned region is visible. For Pick Up Chips, success requires lifting the potato chip fully clear of its support and holding it without visible breakage or dropping it; the support and its base must remain on the table. Figure 6(b) reports the task completion counts.

# F Prompts and Tool Interface

## F.1 Operator prompts

The retained execution prompts are supplied in `appendix_materials/`. The three shared blocks below replace only the task name, goal, static rules, and control-step limit with named fields; task-specific blocks provide their values. Tool argument schemas follow in Section F.3.

**Shared operator prompt: interaction**

You control one live UniVTAC {task_name} episode through OpenETA's general-purpose robot tools.

Start by calling review_demonstrations once, then observe the current episode. Use the current camera images, bilateral tactile images when provided, robot state, and execution feedback to decide your motion targets, orientation, and gripper actions.

**Shared operator prompt: interaction (continued)**

After each action, inspect the returned observations. You may probe, retreat, adjust, and retry within this episode. A recoverable tool failure does not end the task.

Use finish_episode to end voluntarily or acknowledge the neutral episode-ended notification. Voluntary ending is final. Native evaluation and automatic stopping run in the background; current task scores and the specific termination reason are not provided. The demonstration tool may provide historical expert experience or no examples. Use any examples as references for reasoning about the current observations, not as absolute-coordinate scripts. Recorded measured movement is not necessarily a recorded tool command. Historical success does not guarantee current success.

Do not infer task success merely because a command completed.

**Shared operator prompt: coordinates and observations**

Task goal: {task_goal}

Coordinates: xyz_m is the absolute TCP position in world metres; delta_mm is displacement in world or current grip_site axes, in millimetres. The TCP is UniVTAC's measured gripper center (fixed transform from panda_hand), not the LIBERO grip site. approach_world sets TCP +Z; jaw_world sets TCP +X. These fields specify orientation axes, not a task-specific rotation axis.

Unspecified position/orientation/gripper remain unchanged. An omitted gripper preserves the previous two finger actuator targets, including the commands inherited from reset. Explicit open/close changes and retains them. Arm arrival and completion of a finite gripper wait are separate feedback; neither establishes a stable grasp or task success.

After move_to, head/wrist show the current scene and the two tactile image strips show, when provided, real segment-end tactile history on a shared time axis, at up to four times. These are selected image/motion changes, not slip labels or early warnings. The takeover reference is not necessarily a no-contact baseline.

Use preview=true for target resolution without physics. Use execute_preview_id alone to execute that preview. A preview is valid until the next real request.

**Shared operator prompt: budgets and task requirements**

Shared episode budgets and termination semantics: At most 30 admitted non-preview move_to requests, 100 MCP calls and 3600 seconds of total Codex wall time are available. Planning rejection of an admitted motion request consumes its request budget. Native control is limited to {native_control_step_limit} steps, with at most 80 control steps per motion segment. Each control step advances two physics steps at 120 Hz; physics pauses while you reason. A segment limit need not end the episode: inspect the actual feedback. Native terminal or an episode budget ends physical interaction. After native terminal, last-observation and completion tools remain available for up to 300 seconds, within the total Codex wall-time limit. Finish naturally when done.

Static native rules: {static_native_rules}

Some observation modalities may be absent. Use only the images and measurements actually returned. Missing tactile input is not a tool failure.

**Insert Tube task fields**

task_name: insert tube

native_control_step_limit: 300

Task goal: Insert the 2.0 cm diameter test tube into the 2.05 cm narrow hole on the inclined surface from the near-center starting position.

Static native rules: the object's pose origin in the hole frame must satisfy $x<0.005$ m, $y<0.005$ m (these are signed comparisons, not absolute values), $z<-0.03$ m, and the object's positive z axis dotted with the hole positive z axis $>0.965$. The absolute change of object z relative to the gripper from the native reference must be $<0.03$ m. That change $>0.03$ m triggers early termination. Equality does not satisfy success or that early-stop test. Object axial position is not TCP displacement. Current hole

**Insert Tube task fields (continued)**

transform and grip-relative drift are not supplied. These are static requirements, not current measurements. Only neutral episode-ended feedback is returned.

**Lift Can task fields**

task_name: lift can

native_control_step_limit: 300

Task goal: Grasp the horizontal cylindrical can on the ground plane and lift it vertically without slippage.

Static native rules: can world pose-origin z $<$0.01 m and absolute dot product of its x axis with world [0,0,1] $>$0.99. With the configured gsmini sensor, maximum native indentation depth $>$8.0 mm causes early termination. Also, absolute in-hand z change from the native reference $>$0.05 m together with absolute dot product of the can z axis with world [0,0,1] $>$0.99 causes early termination. These are the native predicates, not a minimum lift-height rule. Current indentation, in-hand drift and object ground truth are not supplied. These are static requirements, not current measurements. Only neutral episode-ended feedback is returned.

**Lift Bottle task fields**

task_name: lift bottle

native_control_step_limit: 500

Task goal: Grasp the bottle resting on the ground plane and lift it vertically, keeping the bottle base within 5 cm of the wall at the final pose.

Static native rules: bottle origin in the native target frame must satisfy x$>$-0.02 m, $|$y$|<$0.1 m and $|$z$|<$0.001 m; the absolute dot product of its x axis expressed in that frame with [0,0,1] must be $>$0.99. After more than 300 native action steps, absolute dot product of that x axis with [-1,0,0] $>$0.99 causes early termination. These are target-relative pose comparisons, not a minimum lift-height rule. The current target transform is not supplied. These are static requirements, not current measurements. Only neutral episode-ended feedback is returned.

**Pull Out Key task fields**

task_name: pull out key

native_control_step_limit: 300

Task goal: Rotate the key in the slot until mechanical resistance is sensed, then pull the key straight out.

Static native rules: key world z $>$0.09 m, key positive z axis dotted with world positive z $>$0.965, and slot x axis expressed relative to its native initial reference dotted with [1,0,0] $>$0.99. The absolute difference between native robot-base-relative end-effector z and key world z must be $<$0.14 m. That difference $\geq$0.14 m or the slot-axis dot product $<$0.99 triggers early termination. These are the native comparisons; current key/slot ground truth and these measured values are not supplied. These are static requirements, not current measurements. Only neutral episode-ended feedback is returned.

### F.2 TOOL INTERFACE

Table 9 summarizes the exposed tools. The online TCP is the measured gripper center, offset by 0.131 m along the local hand axis from `panda_hand`; online state quaternions use `xyzw` order. The argument conventions below therefore differ from the historical base-relative hand poses. Omitted pose components and gripper targets remain unchanged. A preview resolves a target without advancing physics; executing it uses its returned identifier, which expires on the next real request.

Table 9: Robot interface. Auxiliary tools expose no online native task score.

| Tool | Arguments and purpose |
|---|---|
| review_demonstrations | No arguments; retrieve demonstration context. |
| observe | No arguments; obtain current observations. |
| move_to | Absolute xyz_m in world metres or relative delta_mm in world/current grip_site axes; approach_world and jaw_world specify TCP $+Z$ and $+X$; optional gripper, preview, and preview identifier. |
| mark_point | view, integer pixel u, v, optional point_id; identify an image point. |
| report_issue | Required message; report an issue. |
| finish_episode | Optional reason; end voluntarily or acknowledge termination. |

### F.3 Argument schemas

The core motion schema below reproduces the retained argument definition, omitting only display-only title fields. Auxiliary tool arguments are summarized in Table 9. The complete file is included as appendix_materials/mcp_tools.json. The protocol declares structured_output=false. Array lengths are not constrained by these schemas; the three-axis interpretation is specified in the operator prompt.

**move_to argument schema**


```
{
"properties": {
"xyz_m": {"anyOf": [{"items": {"type": "number"}, "type":
"array"}, {"type": "null"}], "default": null},
"delta_mm": {"anyOf": [{"items": {"type": "number"}, "type":
"array"}, {"type": "null"}], "default": null},
"delta_frame": {"default": "world", "enum": ["world",
"grip_site"], "type": "string"},
"approach_world": {"anyOf": [{"items": {"type": "number"},
"type": "array"}, {"type": "null"}], "default": null},
"jaw_world": {"anyOf": [{"items": {"type": "number"}, "type":
"array"}, {"type": "null"}], "default": null},
"gripper": {"anyOf": [{"enum": ["open", "close"], "type":
"string"}, {"type": "null"}], "default": null},
"preview": {"default": false, "type": "boolean"},
"execute_preview_id": {"anyOf": [{"type": "string"}, {"type":
"null"}], "default": null}
},
"type": "object"
}
```