

GPT-6 Astra Is a Native Visuo-Tactile Policy for Contact-Rich Manipulation

Chen Yang^{1,2} Linzhe Shi² Hang Zhang² Mu Xu² Jiansheng Fan¹ Chen Wang^{1,†}

¹Tsinghua University ²Amap, Alibaba Group

[†]Corresponding author

Recent advances in multimodal models have enabled GPT-6 Astra to perform robotic manipulation through in-context learning. Yet contact-rich control and tactile perception are often regarded as strengths of specialist policies, leaving open whether Astra can directly understand and use tactile observations. We provide, to our knowledge, the first demonstration that Astra directly interprets vision-based tactile images and uses their contact cues for manipulation. Tactile RGB images enter through its existing visual interface, enabling few-shot closed-loop control on nineteen simulated UniVTAC and OpenViTac tasks with 210 sampled scene seeds per setting. With visuo-tactile input, Astra achieves mean success rates of 93.8% and 81.8%, exceeding the strongest reported specialist baselines in our comparison by 22.9 and 13.1 percentage points, respectively. Vision-only performance also leads these baselines, reaching 76.3% and 80.6%. Trajectory analysis reveals active probing, cautious grasping, contact-guided alignment, and recovery, alongside difficulties with tangential loading and corrective execution. Our findings challenge the exclusivity of specialist policies in contact-rich manipulation and show that fine-grained vision can partially compensate for missing touch. They also identify vision-based tactile sensing as an interface for scaling contact understanding with foundation models and reveal signs of benchmark saturation. These results motivate joint visual-tactile pretraining, harder contact-rich evaluations, and physical-robot validation.

Date: October 20, 2026

Code: <https://github.com/ccyydd/TacTalent>

Project Page: <https://tactalent.github.io/>

Video Gallery: <https://tactalent.github.io/gallery.html>

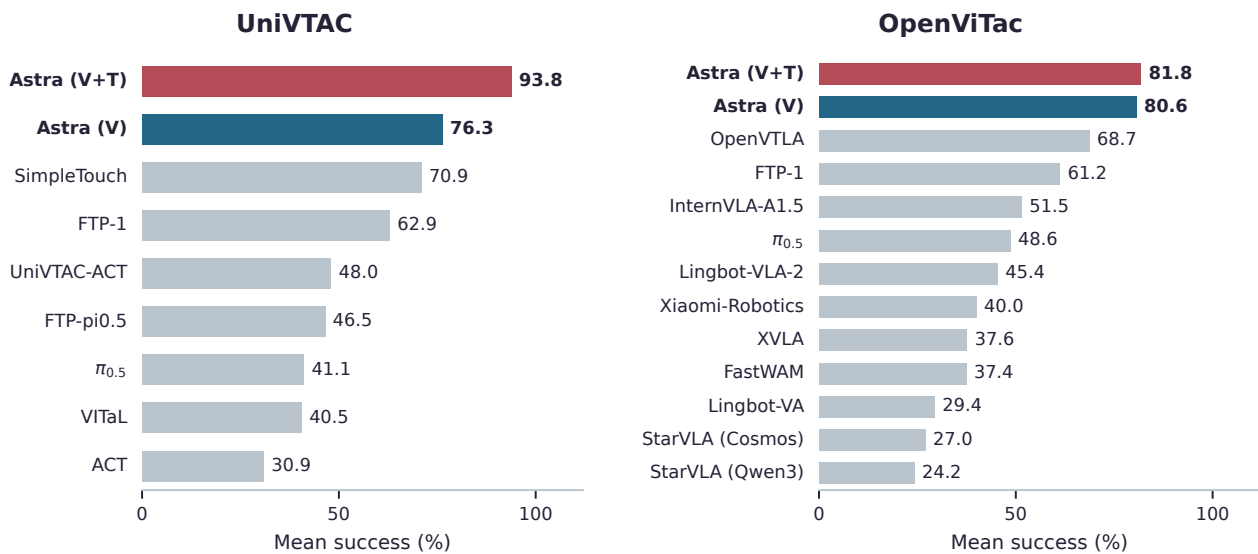


Figure 1. Benchmark-level results. Task-averaged success rates (%). Astra uses vision alone (blue) or vision with tactile RGB images (red). Grey bars show published specialist baselines. Models are ordered by their reported mean within each benchmark. Task-level results and evaluation scope appear in Section 5.

1 Introduction

Recent studies show that GPT-6 Astra can translate visual observations and instructions into robot actions, challenging the assumption that effective manipulation requires a separately trained policy. GPT-as-Policy (Su et al., 2026), RoboProbe (Zhang et al., 2026), RoboDawn (Guo et al., 2026), and RoboICL (Liu et al., 2026) explore this capability through different control interfaces and forms of contextual feedback. GPT-Policy and broader embodied evaluations further examine how demonstration following and task decisions translate into physical execution (Cheng et al., 2026; Galbot Team et al., 2026). These advances shift the question from whether general models can control robots to which capabilities still distinguish specialist policies.

Contact-rich manipulation is a prominent candidate. Precise insertion, stable grasping, and material discrimination depend on local geometry, slip, and physical properties that scene images may not reveal. Motions that work in free space can fail after contact, while a stronger grasp can damage a fragile object. These demands motivate tactile pretraining and learned visual–tactile integration, including FTP-1 (Yuan et al., 2026), OpenVTLA (Wu et al., 2026), and SimpleTouch (Yang et al., 2026a). Can Astra directly use contact evidence for these decisions, or do such interactions require specialist training?

Vision-based tactile sensing makes this question especially relevant. Sensors such as GelSight encode contact geometry and deformation as images (Yuan et al., 2017), exposing touch through the same interface that supports general visual understanding. Few-shot demonstrations relate local deformation to preceding actions, grasp conditions, and resulting object motion. We explore whether Astra can interpret these relationships and use tactile evidence to guide a new interaction, extending visual competence to contact understanding.

We investigate Astra’s direct tactile-image understanding on UniVTAC (Chen et al., 2026a) and OpenViTac (Wu et al., 2026). Tactile RGB images enter through the model’s existing visual interface, allowing us to compare vision-only and visuo-tactile control on the same sampled scenes. We connect task outcomes to recorded contact inference, corrective actions, and recovery. Our contributions are threefold:

1. **Direct tactile-image understanding and control.** To our knowledge, we provide the first demonstration of Astra directly interpreting vision-based tactile images and using their contact cues for manipulation. We investigate this capability on nineteen tasks and 210 sampled scene seeds per input setting.
2. **Leading contact-rich manipulation results.** Astra reaches 93.8% and 81.8% mean success, exceeding the strongest reported specialist means in our comparison by 22.9 and 13.1 percentage points (Figure 1). It outperforms FTP-1 and SimpleTouch on UniVTAC and OpenVTLA on OpenViTac; vision-only performance also leads these baselines at 76.3% and 80.6%.
3. **An analysis of contact understanding and visual compensation.** We connect task-level outcomes to contact-guided probing, fine adjustment, and recovery, and identify subtle visual cues that compensate for missing touch. This analysis reveals both Astra’s tactile competence and its remaining perceptual and control limitations.

Together, these results challenge the presumed moat of specialist contact-rich policies and connect tactile sensing to the development of more capable general models.

2 Related Work

Multimodal Models as Robot Policies. Code as Policies and VoxPoser use general models to generate robot programs and spatial objectives (Liang et al., 2023; Huang et al., 2023). Recent Astra-based work brings the model closer to direct control: GPT-as-Policy compares direct and hybrid policies (Su et al., 2026); RoboProbe and systematic evaluations examine manipulation and execution limits (Zhang et al., 2026; Galbot Team et al., 2026); and RoboDawn and GPT-Policy ground control in demonstrations (Guo et al., 2026; Cheng et al., 2026). RoboICL unifies demonstrations and online experience through an observation–action–receipt–observation grammar, preserving distributed interactions alongside recent feedback (Liu et al., 2026). Tactile feedback has also been studied in Astra-based peg insertion (Tanaka,

2026). This line complements learned action-chunking, diffusion, and pretrained VLA policies (Zhao et al., 2023; Chi et al., 2023; Kim et al., 2025; Physical Intelligence et al., 2025). Our study centers on Astra’s direct interpretation of tactile RGB images and their role in contact-rich control.

End-to-End Policies with Tactile Feedback. VITaL, 3D-ViTac, and ViTacFormer learn visual–tactile representations for manipulation (George et al., 2025; Huang et al., 2024; Heng et al., 2025). Tactile-VLA, VLA-Touch, and AT-VLA incorporate tactile feedback into pretrained VLAs (Huang et al., 2025; Bi et al., 2025; Li et al., 2026). FTP-1 pretrains tactile policies across sensors (Yuan et al., 2026); OpenVTLa studies tactile representations and integration within OpenViTac (Wu et al., 2026); and SimpleTouch adds a tactile expert with predictive contact features, without additional tactile policy pretraining (Yang et al., 2026a). TacDyn-WAM models implicit tactile dynamics by predicting future representations within a heterogeneous visuo-tactile world action model (Wang et al., 2026). Sparsh, AnyTouch, and T3 provide reusable sensory representations across sensing configurations and tasks (Higuera et al., 2024; Feng et al., 2025; Zhao et al., 2025). We examine how much tactile competence a general model can access directly through its visual interface, using demonstrations that relate contact images to actions and outcomes.

Contact-Rich Manipulation Benchmarks. LIBERO, ManiSkill3, and RoboDojo evaluate manipulation learning, transfer, and generalist control (Liu et al., 2023; Tao et al., 2025; Chen et al., 2026b); Bench2Dex extends evaluation to visuo-tactile bimanual manipulation across dexterous hands (Yang et al., 2026b). UniVTAC unifies tactile data generation, learning, and benchmarking (Chen et al., 2026a), supporting established comparisons across grasping, classification, placement, and insertion (Yuan et al., 2026; Yang et al., 2026a). The more recent OpenViTac provides a complementary sim-and-real framework for property discrimination and precise contact manipulation (Wu et al., 2026). Our evaluation spans both suites and examines whether texture, contour, and visible slip can support nominally tactile judgments, helping distinguish fine-grained visual competence from understanding of genuinely unobservable physical properties.

3 Method: Direct Tactile-Image Control

We present tactile images to Astra as direct visual evidence for contact-rich manipulation. Together with scene images and robot state, they expose local deformation and contact geometry through the model’s existing visual interface. Demonstrations connect these observations to actions and outcomes, and online feedback supports contact-guided corrections. Figure 2 shows this interface with an actual Insert Hole observation. Demonstration-conditioned execution and interaction memory follow RoboICL (Liu et al., 2026).

3.1 Scene and Contact Observations

At decision round t , the model receives task instruction g , robot state x_t , scene images I_t , and, in the visuo-tactile setting, the two finger-contact images T_t . Scene images describe global object pose and target geometry; tactile images describe local deformation and contact. The input preserves the benchmark’s configured cameras, including a wrist view where available. Robot state provides the current end-effector pose and proprioceptive information needed to relate an observed contact to a proposed motion. Demonstration and online observations use the same image interface, allowing Astra to compare contacts in relation to their associated actions and outcomes. Vision-only control omits tactile images from both sources.

3.2 Demonstrations and Interaction Memory

The demonstration context $\mathcal{D}$ consists of selected blocks from successful task trajectories. A block contains its initial observation and robot state, the recorded action sequence, an execution receipt, and the resulting observation and state. Blocks are selected before evaluation and remain available throughout an episode. They expose local state transitions across the trajectory; tasks with distinct classes or shapes include demonstrations that cover those branches. Our preparation uses temporal targets constrained by action validity and non-overlap, as detailed in Appendix A.

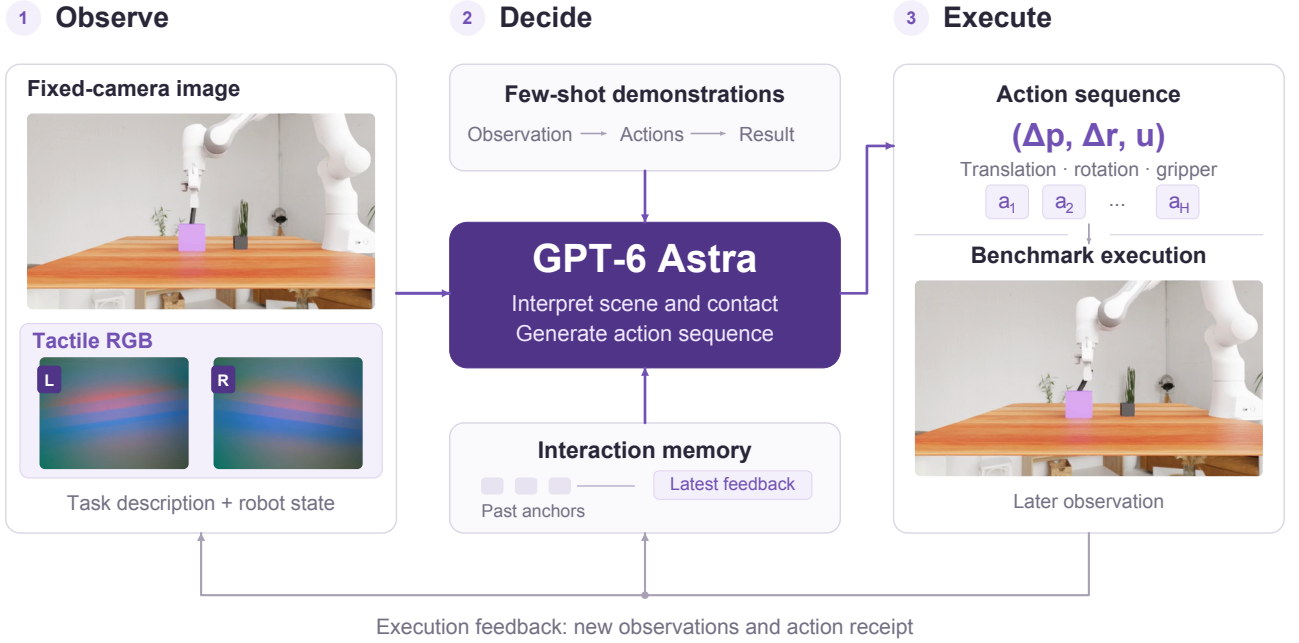


Figure 2. Direct tactile-image input for contact-guided manipulation. Tactile RGB images and scene observations enter Astra through its visual input interface. Demonstrations and online interaction memory connect contact evidence to actions and outcomes. Astra generates end-effector and gripper commands; benchmark execution returns observations and a receipt for the next decision. The displayed fixed-camera and tactile images are actual inputs from Insert Hole, seed 1000028. Vision-only evaluation omits the tactile images throughout the context.

Online memory $\mathcal{M}_t$ uses the same interaction structure. Distributed anchors retain earlier task stages, while the latest interaction exposes the most recent execution result. Explicit gap markers identify omitted intervals. This bounded memory relates the current observation to earlier attempts and demonstrated procedures (Liu et al., 2026).

3.3 Action Generation and Feedback

Astra produces an ordered sequence of end-effector and gripper commands from the assembled context:

$$A_t \sim \pi(\cdot \mid g, \mathcal{D}, \mathcal{M}_t, x_t, I_t, T_t), \quad a = (\Delta p, \Delta r, u), \quad (1)$$

where Δp and Δr specify translation and rotation increments and u is the gripper target. In the vision-only setting, T_t is absent. The benchmark adapter converts these commands to the robot’s control interface and records their execution. The receipt and next observation complete the interaction used for the following decision.

Execution feedback distinguishes a completed command from achieved task progress: an insertion command can still leave an object blocked at the rim. New observations allow Astra to revise the next action. Short action notes accompany decisions and link commands to the model’s assessment; our analysis combines these notes with the visible changes and task outcomes.

4 Experimental Setup

Benchmarks and Sampling. We evaluate eight UniVTAC tasks and eleven OpenViTac tasks (Chen et al., 2026a; Wu et al., 2026). UniVTAC supplies an established tactile-policy comparison across grasping, placement, classification, and precise insertion. The more recent OpenViTac adds property discrimination and contact-sensitive assembly within a paired sim-and-real framework. Across seven policies on eight shared tasks, its authors report strong agreement between simulated and real-world performance (Pearson $r = 0.967$), supporting its use for comparative evaluation (Wu et al., 2026). We use a 10% stratified sample, with sampling seed 42, preserving class and layout proportions. Each input setting contains 80 UniVTAC and 130 OpenViTac scene seeds: ten per task, except Insert Block v2, which contains ten per shape across three shapes.

Model and Demonstrations. The model is GPT-6 Astra with xhigh reasoning effort. We use UniVTAC’s released training recordings (Chen et al., 2026a) and collect successful OpenViTac demonstrations with its official scripted experts in training mode (Wu et al., 2026). Demonstrations provide context, while separate evaluation episodes determine success rates. Twelve tasks use one demonstration with twelve selected segments. Six classification or selection tasks use two demonstrations with five segments each; Insert Block v2 uses three demonstrations, one per shape, also with five segments each. Segment length H is chosen with reference to trajectory length and fixed per task. Each segment pairs its H actions with the initial and resulting observations. Appendix A gives every task’s configuration and the selection procedure. All selected demonstration blocks remain available at every decision throughout the episode.

Input Conditions and Camera Access. Vision-only (V) and visuo-tactile (V+T) settings use the same demonstration trajectories, segment starts, actions, and sampled evaluation scenes. V excludes tactile images from both demonstrations and online observations; V+T additionally includes the left and right tactile RGB images. Camera selection strictly follows each benchmark’s official task configuration. UniVTAC uses only the fixed camera in six tasks and also provides the wrist camera for Lift Can and Insert Tube. OpenViTac provides both fixed and wrist cameras for every task. All settings also receive task text and robot proprioception, and generate the seven-dimensional commands in Equation 1.

Success Criteria. All tasks retain their official success criteria except Grasp Chip. Its original conditions impose unnecessarily strict execution details beyond intact object transfer. We simplify these conditions: an intact chip must be grasped, transferred, and placed with stable physical support in the target region. The adjusted criterion accepts supported placement without requiring a particular release-and-withdrawal sequence. The task still requires maintaining the chip’s integrity during manipulation.

Baselines and Metrics. The tables compare Astra with published benchmark results, including action-chunking, VLA, world-action, and tactile-policy baselines (Zhao et al., 2023; George et al., 2025; Physical Intelligence et al., 2025; Yuan et al., 2026; Yang et al., 2026a; Wu et al., 2026). Astra is evaluated on the sampled scenes; baseline values are the corresponding reported benchmark results. We report per-task success percentages and equal-weight means across the eight or eleven tasks. The thirty Insert Block v2 scenes are aggregated into a single task score with the same weight as every other task. The task breakdown separately examines property judgments, constrained insertion, stable grasping, and transfer.

5 Results: Five Findings

5.1 Leading Performance on Contact-Rich Tasks

With visual and tactile inputs, Astra reaches 93.8% mean success on UniVTAC and 81.8% on OpenViTac (Tables 1 and 2). The strongest reported baseline means in these tables are 70.9% for SimpleTouch (Yang et al., 2026a) and 68.7% for OpenVTLA (Wu et al., 2026), yielding gaps of 22.9 and 13.1 percentage points. Vision-only Astra also exceeds these means, reaching 76.3% and 80.6%, respectively. These results establish general multimodal models as strong contact-rich manipulation policies, surpassing the reported specialist baselines in both benchmark comparisons.

Astra also exceeds FTP-1 (Yuan et al., 2026) by 30.9 percentage points on UniVTAC and 20.6 points on OpenViTac. The improvements extend to precise insertion and fragile transfer, complementing the semantic and open-ended strengths identified by earlier Astra evaluations (Zhang et al., 2026; Liu et al., 2026). Weight Classify remains difficult, while several classification, transfer, and insertion tasks reach 100% in the sampled episodes.

The task breakdown reveals that the gains span more than one operation family. On UniVTAC, the largest changes occur in constrained extraction and hole insertion, while the already strong Lift Bottle and Grasp Classify results remain unchanged. On OpenViTac, improvements in Hardness Classify and Insert Block v1 coexist with high vision-only scores on texture discrimination and several transfer tasks. This pattern motivates examining what evidence drives a particular decision, rather than interpreting touch as a uniform addition to every task.

Table 1. UniVTAC success rates (%). Baseline values are reported in UniVTAC and SimpleTouch (Chen et al., 2026a; Yang et al., 2026a). Astra uses ten sampled scene seeds per task. V: vision; V+T: vision and tactile RGB. Avg. is the unweighted mean over tasks. Bold denotes the best value in each column, including ties.

Model	Lift Bottle	Pull-out Key	Lift Can	Put Bottle	Insert Hole	Insert HDMI	Insert Tube	Grasp Classify	Avg.
ACT	42	28	20	28	19	15	45	50	30.9
VITaL	72	47	8	32	25	6	34	100	40.5
$\pi_{0.5}$	92	36	59	31	26	3	34	48	41.1
FTP-pi0.5	77	30	26	19	47	4	72	97	46.5
UniVTAC-ACT	71	46	29	31	24	28	56	99	48.0
FTP-1	97	48	65	47	64	3	79	100	62.9
SimpleTouch	100	62	68	50	85	3	100	99	70.9
Astra (V)	100	30	80	90	60	50	100	100	76.3
Astra (V+T)	100	90	100	90	100	80	90	100	93.8

Table 2. OpenViTac success rates (%). Baseline values are reported in OpenViTac (Wu et al., 2026). Astra uses ten sampled scene seeds per task, except Insert Block v2 (thirty across three shapes). V: vision; V+T: vision and tactile RGB. Avg. is the unweighted mean over tasks. Bold denotes the best value in each column, including ties. Class. abbreviates Classify; Roughness Regrasp abbreviates Roughness-Guided Regrasp. Grasp Chip follows the adjusted criterion described above.

Model	Weight Class.	Hardness Class.	Roughness Class.	Roughness Regrasp	Empty-Can Selection	Grasp Chip	Gear Assembly	Pull Drawer	Insert USB	Insert Block v1	Insert Block v2	Avg.
StarVLA (Qwen3)	44	38	42	35	32	10	26	35	4	0	0	24.2
StarVLA (Cosmos)	51	67	94	9	13	1	33	22	3	1	3.3	27.0
Lingbot-VA	50	44	39	40	44	22	59	15	2	2	6	29.4
FastWAM	49	84	67	7	9	50	51	88	6	0	0	37.4
XVLA	28	30	56	47	45	42	84	76	2	3	0.7	37.6
Xiaomi-Robotics	52	46	54	62	38	24	73	84	5	2	0	40.0
Lingbot-VLA-2	52	58	58	53	52	41	78	92	8	5	2.7	45.4
$\pi_{0.5}$	53	60	64	52	48	39	87	99	14	9	9.3	48.6
InternVLA-A1.5	53	76	65	77	47	53	79	98	3	6	9.3	51.5
FTP-1	61	100	100	78	50	60	83	94	27	12	8.3	61.2
OpenVTLa	89	99	100	90	73	69	91	99	27	11	8	68.7
Astra (V)	50	70	100	100	70	100	80	100	100	40	76.7	80.6
Astra (V+T)	40	90	100	100	50	100	80	100	100	70	70	81.8

5.2 Direct Tactile Understanding and Control

Adding tactile RGB input improves four UniVTAC tasks: Pull-out Key increases from 30% to 90%, Lift Can from 80% to 100%, Insert Hole from 60% to 100%, and Insert HDMI from 50% to 80%. On OpenViTac, Hardness Classify improves from 70% to 90%, and Insert Block v1 from 40% to 70%. The corresponding successful trajectories illustrate resistance detection, grasp stabilization, contact-guided alignment, and material-property judgment.

Vision-based tactile sensing encodes local contact in images (Yuan et al., 2017) that Astra can process through its existing visual interface. In successful traces, changes in contact shape, symmetry, shear, and stability are associated with targeted adjustments: unloading a constrained object, changing the approach direction, testing retention, or limiting closure after contact. These behaviors demonstrate useful tactile interpretation through the same image interface used for ordinary scene perception. The successful interventions depend on the contact regime. Constrained insertion calls for realignment before renewed advancement; fragile grasping calls for stopping closure once support is established; material classification uses controlled changes in contact to obtain a more informative comparison with demonstrations.

The following eight successful trajectories illustrate representative manipulation behaviors, organized into contact-state inference, active probing, fine adjustment, and manipulation recovery. Each storyboard presents recorded observations and concise action summaries. Step labels are zero-based control-action indices, not model-request counts; selected intervals are nonuniform. Camera crops improve legibility, while tactile images retain their original content and aspect ratio. The analysis combines recorded action

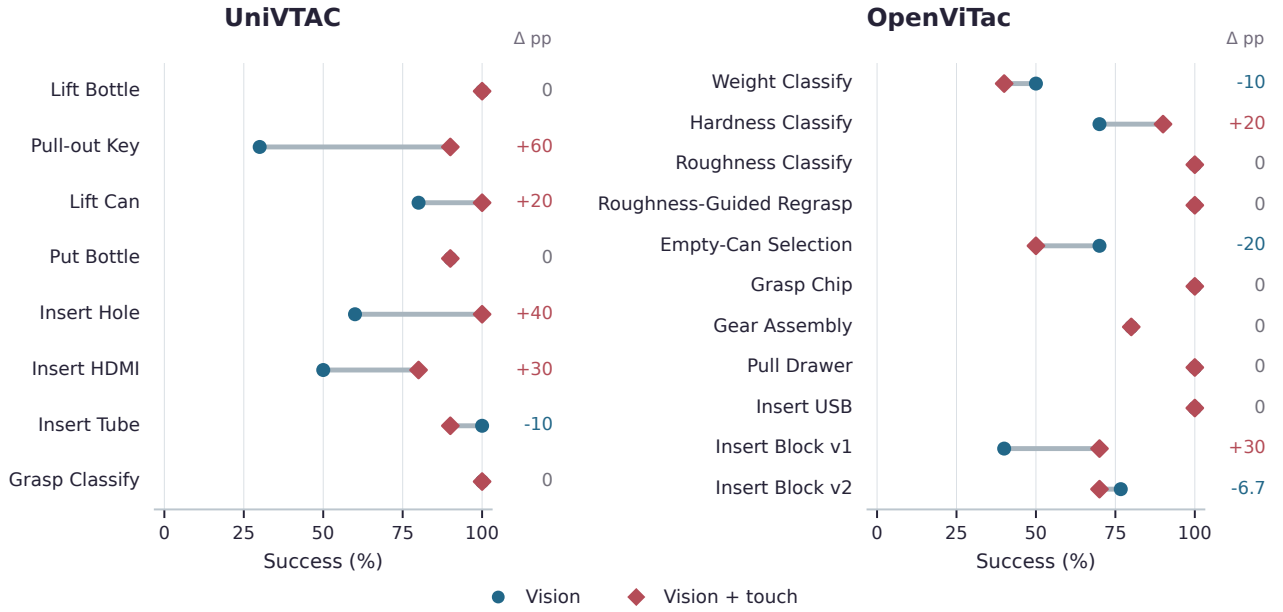


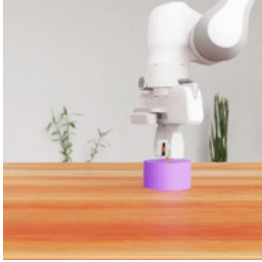
Figure 3. Per-task changes with tactile input. Circles denote V, diamonds denote V+T, and the right-hand numbers show V+T minus V in percentage points. Six tasks improve, nine are unchanged, and four decrease.

notes with visible execution. Action notes describe the policy assessment, and the recorded observations show the resulting motion and task outcome. Appendix B presents same-seed outcomes for the six tasks that improve with touch, connecting the task-level comparison to individual trajectories.

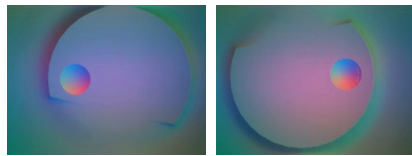
5.2.1. Contact-State Inference

A useful tactile policy must connect changing contact observations to an appropriate intervention. The first two cases illustrate responses to constrained motion and inference about a partially hidden insertion direction.

(a) Step 60



Fixed camera (crop)

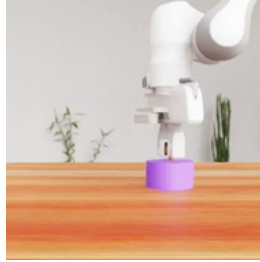


Left tactile

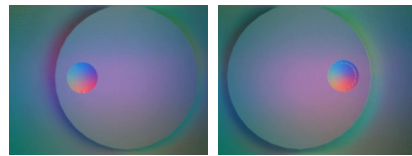
Right tactile

Torsional resistance

(b) Step 80



Fixed camera (crop)

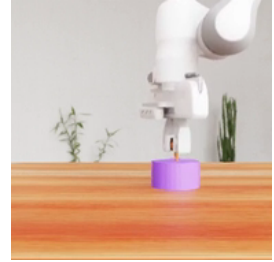


Left tactile

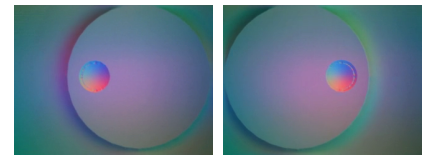
Right tactile

Reseat and refine angle

(c) Step 125



Fixed camera (crop)



Left tactile

Right tactile

Straight extraction

Figure 4. Pull-out Key: resistance-aware extraction. Seed 1000017, V+T, success. At step 60, the action note identifies torsional deformation and proposes unloading the twist. Continued axial resistance leads to reseating and finer angular adjustment at step 80; the later frame shows extraction. The sequence alternates extraction attempts with unloading and realignment as resistance persists. [Full rollout](#).

The informative feature of the Pull-out Key sequence (Figure 4) is the alternation between testing extraction and reducing the constraint that blocks it. Continued resistance changes the next action from another pull to unloading and angular refinement. This matters because larger displacement in the same direction

would not resolve a poorly aligned or twisted contact. The later extraction provides a visible outcome for the revised strategy; the tactile interpretation is useful through the corrective action it supports.

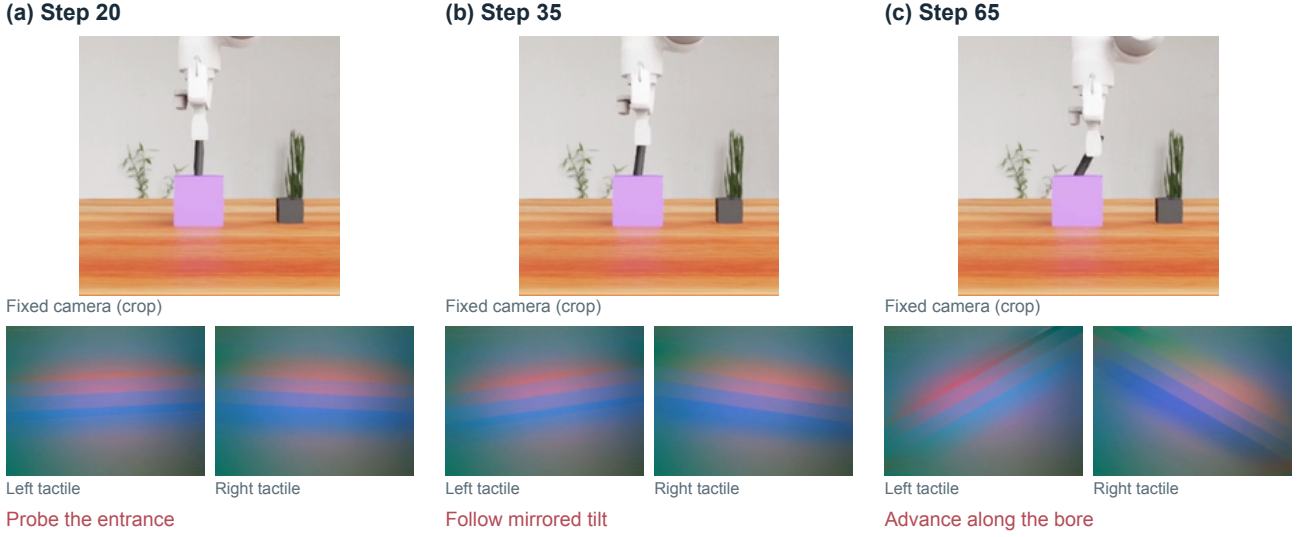


Figure 5. Insert Hole: inferring the bore direction. Seed 1000028, V+T, success. Astra interprets the opposing slopes of the tactile bands together with visible shaft deflection to infer a direction different from the reference trajectory. Astra pivots the shaft while maintaining entrance contact, then advances along the inferred bore. The case illustrates adaptation to contact geometry rather than literal replay of the demonstrated motion. [Full rollout](#).

In Insert Hole (Figure 5), the target direction is partly hidden by the surrounding geometry. The tactile pattern and visible shaft deflection provide a local reference for adjusting the approach. Astra tests its inferred direction through subsequent advancement, adapting the demonstrated procedure to the current contact geometry. The sequence shows how local observations support a new motion direction while preserving the overall task goal.

5.2.2. Active Probing

Astra sometimes acts to obtain a more discriminative observation before committing to a task decision. In Hardness Classify, a change in grip creates an informative response, but also requires recovery when the object begins to slip.

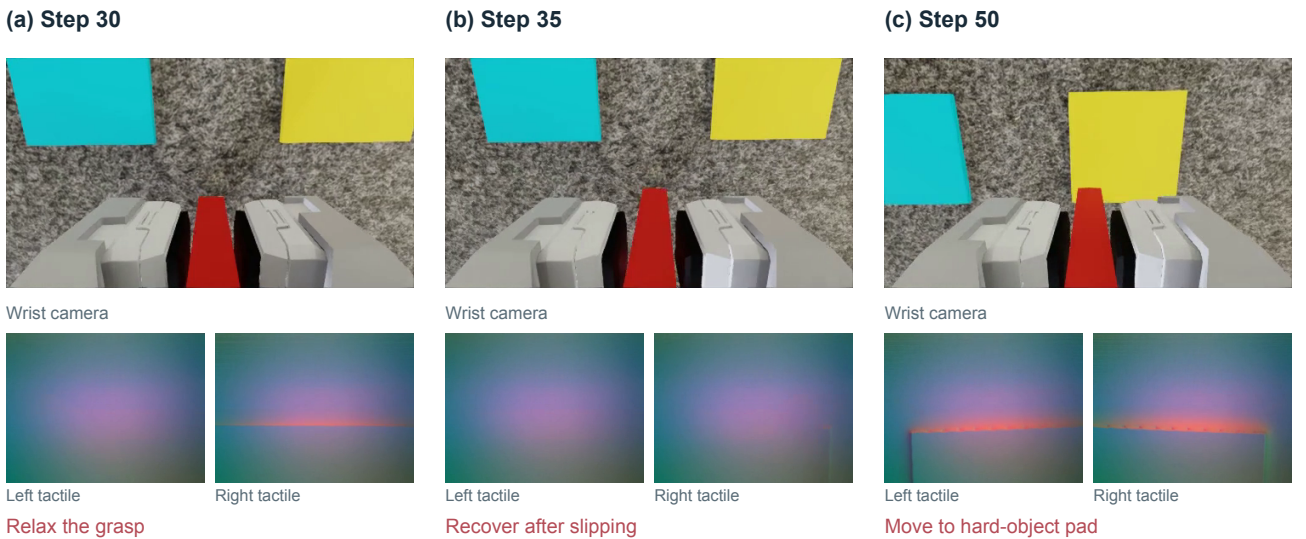


Figure 6. Hardness Classify: probing while maintaining retention. Seed 10006, V+T, success. Astra slightly relaxes the grasp after lifting, detects slipping, and re-establishes contact. The subsequent action note associates the sharply defined loaded edge with the hard-object demonstration. The probe changes the contact condition, and the subsequent correction preserves retention before classification. [Full rollout](#).

The Hardness Classify probe (Figure 6) is also a control intervention: relaxing the fingers changes both the observation and the security of the grasp. The subsequent recovery is therefore part of the probing behavior. Astra must maintain the object while obtaining evidence for its classification. The sequence connects a deliberately changed contact condition to a reference-based property judgment, instead of relying on a single unchanging tactile appearance.

5.2.3. Fine Adjustment

Near contact, small changes can determine whether a grasp succeeds without damaging the object. The Grasp Chip sequence separates first contact, centering, bilateral support, and the retention test.

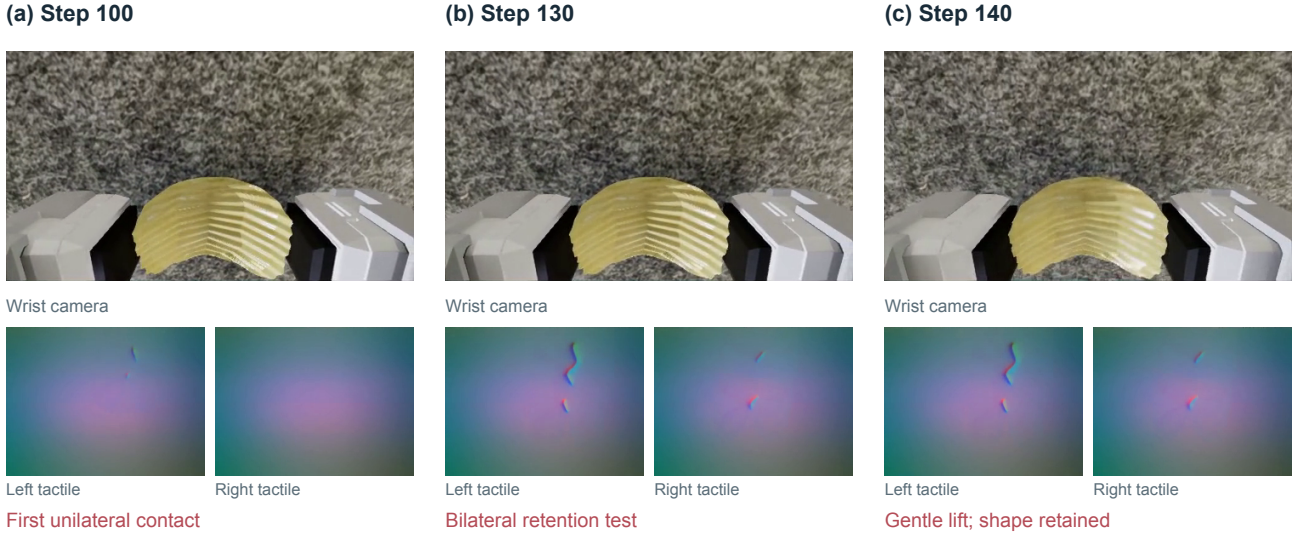


Figure 7. Grasp Chip: cautious convergence on a fragile grasp. Seed 10099, V+T, success. Unilateral contact triggers smaller closure increments and a commanded 0.5 mm lateral correction at step 110, between panels (a) and (b). After bilateral contact develops, Astra avoids continued tightening and tests retention through a gentle lift. The episode later completes intact placement; the panels focus on the contact-establishment phase. [Full rollout](#).

The Grasp Chip sequence (Figure 7) separates establishing contact from increasing grip strength. Initial contact on one side prompts a small positional correction, after which bilateral contact can be checked before lifting. This ordering is useful for a fragile object because continued closure is not the only way to improve retention. A gentle lift then tests whether the current support is sufficient. The successful outcome depends on coordinating these stages rather than issuing one aggressive closing command.

Successful correction requires identifying which part of an operation is blocked. Gear assembly separates center alignment from tooth engagement; USB insertion distinguishes hand displacement from actual connector progress.

The gear in Figure 8 presents successive constraints at different stages of assembly. Recentering can address a tilted approach, but it does not by itself establish a compatible tooth configuration. The later angular change reflects this distinction. The observed behavior is therefore more specific than a general instruction to move closer to the target: the policy revises the type of correction as the operation progresses from gross alignment toward engagement and seating.

The contrast between hand motion and object progress is central to the USB case (Figure 9). A moving gripper can change the tactile contact even when the plug remains blocked. The exposed connector and shifting finger contacts help distinguish these possibilities. Withdrawing and recentering changes the contact configuration before another advance. This is a useful form of progress checking because issuing a displacement command alone does not establish that the intended insertion has occurred.

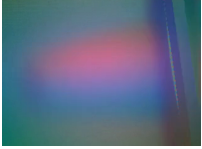
Grasp preparation and local correction can both influence insertion. The successful Insert Block v1 case combines an initial orientation adjustment with a small retreat and recentering near the opening.

In Insert Block v1 (Figure 10), the initial grasp is part of the alignment problem: the chosen hand orientation influences how the cube is presented to the opening. Near the rim, the policy uses a different

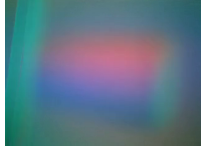
(a) Step 90



Wrist camera



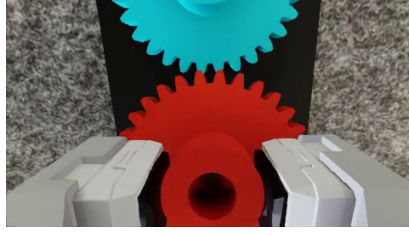
Left tactile



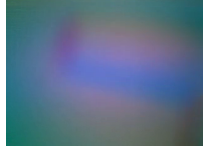
Right tactile

Unload tilted gear

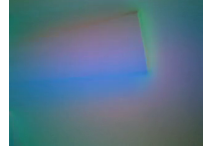
(b) Step 140



Wrist camera



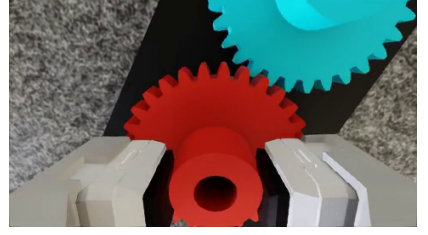
Left tactile



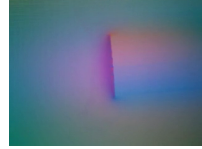
Right tactile

Refine tooth engagement

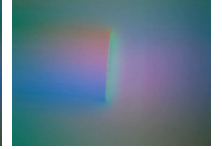
(c) Step 190



Wrist camera



Left tactile

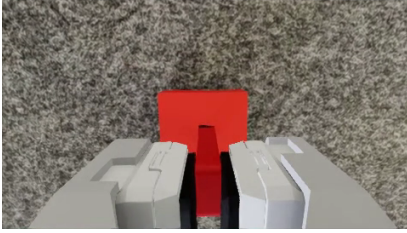


Right tactile

Rotate the seated gear

Figure 8. Gear Assembly: resolving successive assembly constraints. Seed 10034, V+T, success. When the red gear tilts, Astra unloads contact and adjusts its center. A later yaw adjustment addresses possible tooth interference before seating and rotation. The correction strategy changes from centering to tooth engagement as the observed constraint changes. [Full rollout](#).

(a) Step 45



Wrist camera



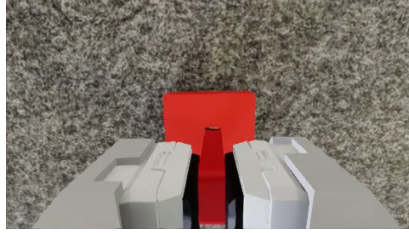
Left tactile



Right tactile

Contact shifts; tip exposed

(b) Step 55



Wrist camera



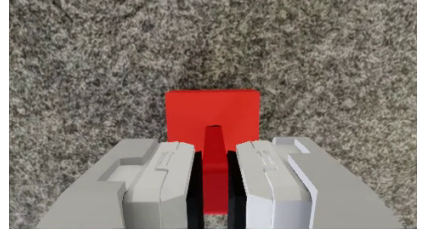
Left tactile



Right tactile

Withdraw the blocked plug

(c) Step 80



Wrist camera



Left tactile



Right tactile

Controlled insertion

Figure 9. Insert USB: verifying object progress. Seed 10066, V+T, success. The connector remains exposed while the tactile contacts shift, and the action note identifies finger slip along a blocked plug. Astra withdraws, recenters, and issues a submillimeter correction before insertion progresses. Hand motion is not treated as sufficient evidence that the connector has entered the socket. [Full rollout](#).

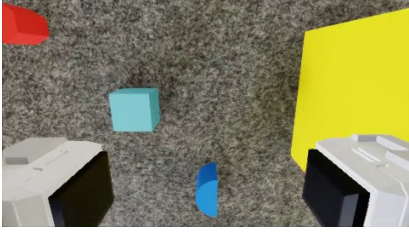
scale of correction, combining a small retreat with millimeter-scale recentering. The retreat creates an opportunity to change the approach before renewed entry. The episode links preparation at a distance with local adjustment near contact, while the final release checks the resulting placement.

5.2.4. Manipulation Recovery

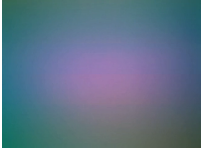
A release command does not establish that an insertion has succeeded. The next case checks the resulting object state and reopens the manipulation sequence when the part remains on the rim.

Recovery in Insert Block v2 (Figure 11) begins with a discrepancy between the intended and observed result. The release action has finished, but the object has not reached the required state. Astra consequently treats the task as unfinished and initiates another grasp-and-alignment sequence. This distinction between command completion and task completion is essential in contact-rich manipulation, where a plausible action sequence can leave an object balanced on an edge or obstructed at an entrance.

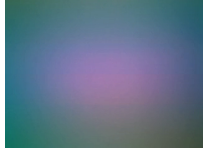
(a) Step 10



Wrist camera



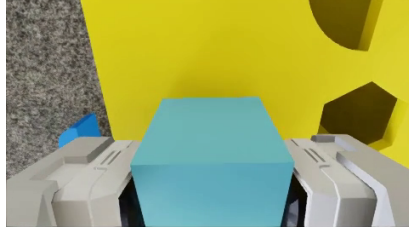
Left tactile



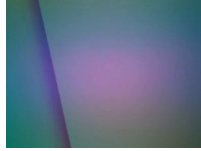
Right tactile

Prepare grasp orientation

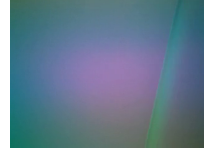
(b) Step 100



Wrist camera



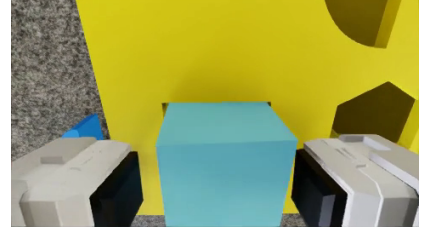
Left tactile



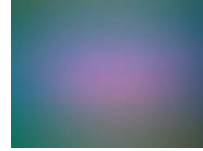
Right tactile

Back off and recenter

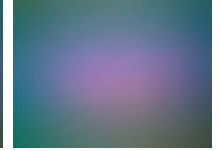
(c) Step 120



Wrist camera



Left tactile

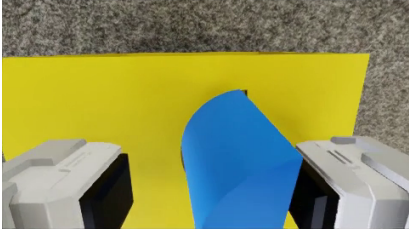


Right tactile

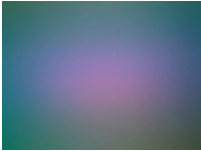
Release and verify entry

Figure 10. Insert Block v1: a local correction after rim contact. Seed 10036, cube, V+T, success. Astra prepares the hand orientation before grasping, transports the cube, and then responds to possible rim contact with a small retreat and a commanded 2 mm centering adjustment. The final frame shows the release-and-verification phase. The local retreat creates clearance for recentering before renewed insertion. [Full rollout](#).

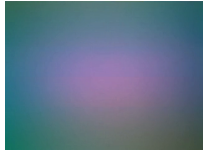
(a) Step 100



Wrist camera



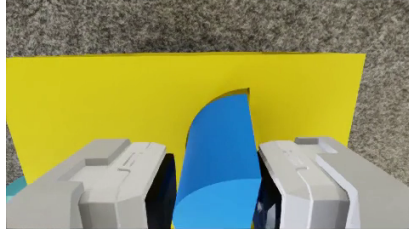
Left tactile



Right tactile

Released part stays on rim

(b) Step 120



Wrist camera



Left tactile



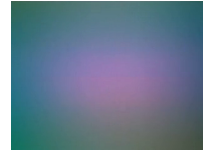
Right tactile

Regrasp and lift clear

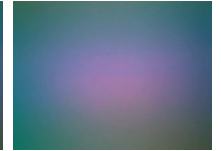
(c) Step 218



Wrist camera



Left tactile



Right tactile

Part settles into opening

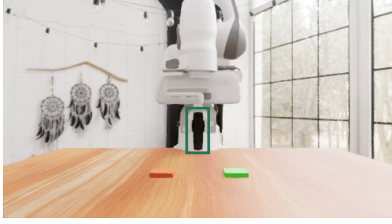
Figure 11. Insert Block v2: recovering from unsuccessful placement. Seed 10019, half-cylinder, V+T, success. After release, the part remains at the opening. Astra regrasps, lifts, and makes further alignment attempts until the part settles into the opening. The behavior demonstrates outcome verification and recovery from an incomplete placement. [Full rollout](#).

5.3 Visual Compensation for Missing Touch

The gain associated with tactile input differs substantially between benchmarks: 17.5 percentage points on UniVTAC and 1.2 points on OpenViTac. Their camera configurations provide one possible explanation. UniVTAC restricts some visual cues in its tactile evaluation setting: six of its eight tasks use only the fixed camera, while Lift Can and Insert Tube also use a wrist camera. OpenViTac provides both fixed and wrist cameras across its tasks, yielding richer visual observations. These differences may reduce the marginal value of touch for a model that can identify subtle visual structure.

In all three tasks in Figure 12, Astra identifies visible texture or ridges in its action notes and uses these cues to complete the required classification or selection without tactile images. This level of pure-vision discrimination goes beyond what the reported specialist results demonstrate. Its sensitivity exposes subtle details that human observers can easily overlook, showing why visually inconspicuous cues still matter when designing tasks intended to require touch.

(a) Grasp Classify



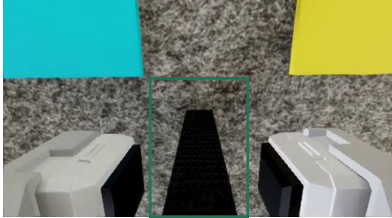
Seed 1000016 | step 0 | Vision



Visible cylinder texture

Fixed-camera crop

(b) Roughness Classify



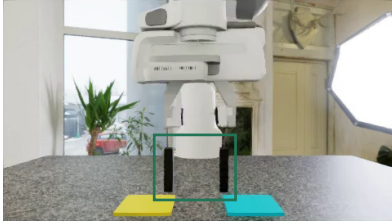
Seed 10025 | step 10 | Vision



Surface ridges and outline

Wrist-camera crop

(c) Roughness-Guided Regrasp



Seed 10002 | step 0 | Vision



Two different surface textures

Fixed-camera crop

Figure 12. Visible cues in nominally tactile tasks. Three V episodes from tasks with 100% sampled success. Left: a recorded camera view; center: enlargement of the green-marked region; right: the visible cue. Grasp Classify shows cylinder texture, Roughness Classify exposes surface ridges and contour structure, and Roughness-Guided Regrasp contains visually different surfaces even in the fixed-camera view. Enlargements preserve the recorded pixels. Rollouts: [Grasp Classify](#), [Roughness Classify](#), and [Roughness-Guided Regrasp](#).

The three cases expose different routes to visual discrimination. Grasp Classify provides texture on the cylinder surface; Roughness Classify adds a close view of ridges and object contour; Roughness-Guided Regrasp presents distinguishable surfaces in the fixed-camera scene. Such cues can support the required category or selection decision even without directly measuring the physical property through touch. Their usefulness depends on the correspondence between visible appearance and the task categories, as well as the model’s ability to notice it.

The visual evidence is not limited to static texture. In three inspected Weight Classify pairs where V succeeds and V+T fails, the vision-only policy relaxes the grasp and uses visible slip to reach the correct category. The tactile runs instead associate weak shear or a contact pattern resembling the light-object demonstration with low weight and misclassify the heavy object. Empty-Can Selection shows a related ambiguity: subtle contact changes can be misread, while repeated comparisons leave insufficient actions for transfer. These observations connect the visual-compensation result to concrete failures in evidence interpretation and probe selection, examined in Section 6.

5.4 Scaling Potential of Vision-Based Touch

The preceding results identify an important property of vision-based tactile sensing: it expresses contact through an interface that a general multimodal model already processes. GelSight-style sensors convert local deformation into images (Yuan et al., 2017); the model can relate these images to object appearance, robot state, preceding motion, and demonstrated outcomes. In our experiments, that relationship supports decisions about constrained motion, retention, and alignment without training a new tactile policy for each task.

This observation complements work on reusable tactile representations. VITaL, Sparsh, AnyTouch, and T3 learn representations that transfer across tactile settings (George et al., 2025; Higuera et al., 2024; Feng et al., 2025; Zhao et al., 2025); FTP-1 and SimpleTouch connect contact-sensitive representations to action generation (Yuan et al., 2026; Yang et al., 2026a). Astra exposes a further route: a single general visual interface can interpret both scene images and contact images through demonstration-conditioned inference. The successful behaviors in Section 5.2 illustrate how temporal context connects contact observations to corrective actions. A contact image becomes actionable when the model connects it to what the fingers did and what happened to the object.

Joint visual–tactile pretraining is a natural next step. Aligned scene images, tactile images, actions, and outcomes could make this correspondence part of general multimodal learning rather than a relationship supplied only at deployment. The resulting research question concerns how contact understanding improves with data diversity, model capacity, and interaction experience. Our results motivate this direction through the capability already accessible from tactile RGB inputs. For sensor development, they favor representations that preserve interpretable contact structure and support comparison across sensors, grasps, and motion conditions.

5.5 Emerging Benchmark Saturation

With tactile input, Astra reaches at least 90% success on thirteen of nineteen tasks, including perfect sampled success on nine. UniVTAC’s mean is 93.8%, while several OpenViTac classification and transfer tasks also reach 100%. These ceilings limit how well the corresponding tasks distinguish stronger policies. They also shift attention from average completion toward the quality of contact decisions, recovery, and generalization.

Visual compensation adds a separate benchmark-design requirement. A tactile task should isolate information that the chosen visual interface does not already reveal. In the current suites, texture, contour, or motion can correlate with the required physical judgment. Astra’s results demonstrate that such cues become useful to a sufficiently capable visual model, even when they are not salient to a human observer. UniVTAC’s restricted camera access and OpenViTac’s richer sim-and-real observation setting produce complementary tests of this issue (Chen et al., 2026a; Wu et al., 2026). Both contact fidelity and control over alternative visual evidence matter when interpreting a tactile-policy result.

More discriminative evaluations should include visually indistinguishable objects with different mass, friction, stiffness, or internal constraints; tighter insertion tolerances; longer contact sequences; and recovery after disturbed grasps or failed engagement. Varying appearance independently of physical properties would test whether a policy infers contact rather than memorizes a visual correlate. Evaluating the same judgments across sensors and real robots would further separate robust tactile understanding from a particular rendering or contact configuration. The broader benchmark goal is reliable physical inference and correction, not simply a larger set of scenes with the same available cues.

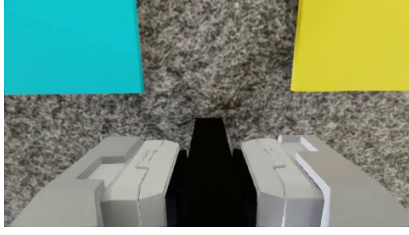
6 Remaining Limitations of Astra

Tactile input does not consistently improve performance: Weight Classify, Empty-Can Selection, Insert Tube, and Insert Block v2 decrease in the sampled results. Three recurring limitations emerge from the trajectories: limited sensitivity to tangential motion and force, difficulty with fine adjustments, and active probing that fails to converge. Weak tactile cues can be misread, a plausible correction can fail to change the object configuration as intended, and repeated observations can leave too little time to finish the manipulation.

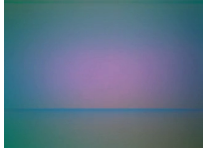
6.1 Limited Sensitivity to Tangential Motion and Force

The ambiguity in Weight Classify (Figure 13) lies in relating the image change to the quantity being inferred. During a lift, the observed contact depends on how the object is supported and whether it moves relative to the fingers. A small change may therefore be insufficient to distinguish low load from a contact configuration that transmits little visible shear. In this episode, Astra commits to the light-object

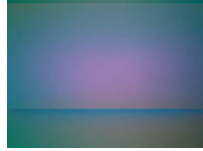
(a) Step 20



Wrist camera



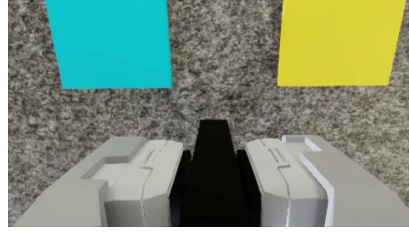
Left tactile



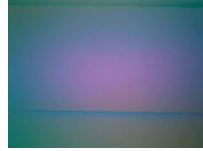
Right tactile

Lift at fixed opening

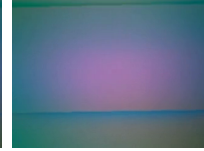
(b) Step 30



Wrist camera



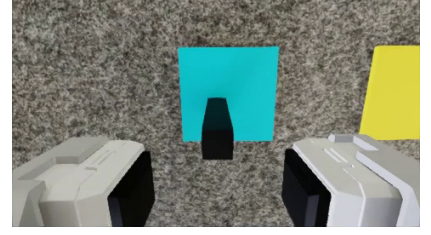
Left tactile



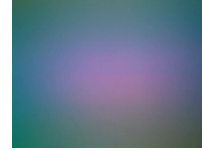
Right tactile

Infer light from weak shear

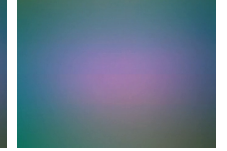
(c) Step 55



Wrist camera



Left tactile



Right tactile

Wrong tray; stable placement

Figure 13. Limited sensitivity to tangential motion and load. Weight Classify, seed 10038, V+T, failure. Astra interprets small tactile changes during a lift as evidence of a light object and places the heavy block on the incorrect blue tray. The image pattern also depends on grasp position, aperture, and motion; a small shear response is therefore not an unambiguous weight measurement. [Full rollout](#).

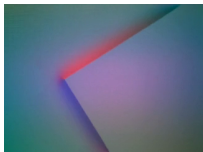
interpretation without resolving that ambiguity, showing a gap between recognizing tactile variation and reliably interpreting tangential loading.

6.2 Fine Adjustments Remain Challenging

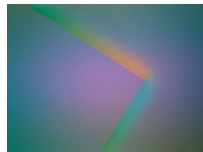
(a) Step 130



Wrist camera



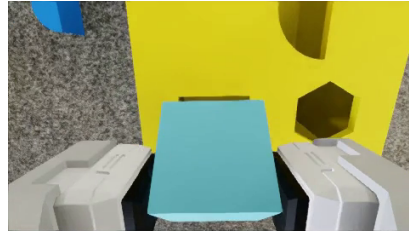
Left tactile



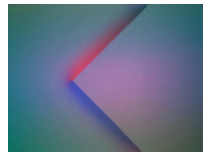
Right tactile

Cube caught on rim

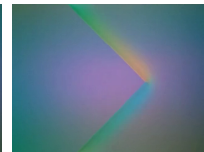
(b) Step 240



Wrist camera



Left tactile



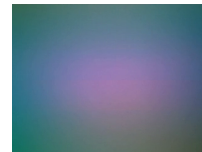
Right tactile

Large leveling command

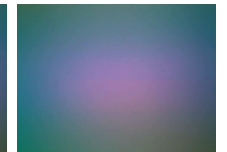
(c) Step 290



Wrist camera



Left tactile



Right tactile

Insertion still incomplete

Figure 14. Recognizing misalignment does not ensure effective correction. Insert Block v1, seed 10016, V+T, failure. Repeated rim contact prompts regrasping and leveling. From step 240 onward, the cumulative commanded pitch increments total approximately 51.6° , with compensatory translation, yet insertion remains incomplete at the action limit. [Full rollout](#).

The Insert Block v1 failure (Figure 14) also separates diagnosing an error from executing a successful correction. A large accumulated rotation command does not specify the object’s actual rotation under contact. Translation, rotation, and the location of support must remain compatible for the part to clear the rim and enter. The repeated attempts show that identifying the need to level the cube is insufficient when the resulting motion does not produce the required object configuration.

Detecting an obstruction does not by itself identify the correction needed to remove it. The failed Insert HDMI episode in Figure 15 makes this distinction explicit. At action 235, Astra states that the tactile image-axis sign is uncertain and tests the opposite pitch direction. At action 270, it returns toward an upright interpretation after the alternate pitch still produces slip. At action 585, the action note again

identifies obstruction rather than seating. The episode ends without success after 600 actions and 120 model decisions.

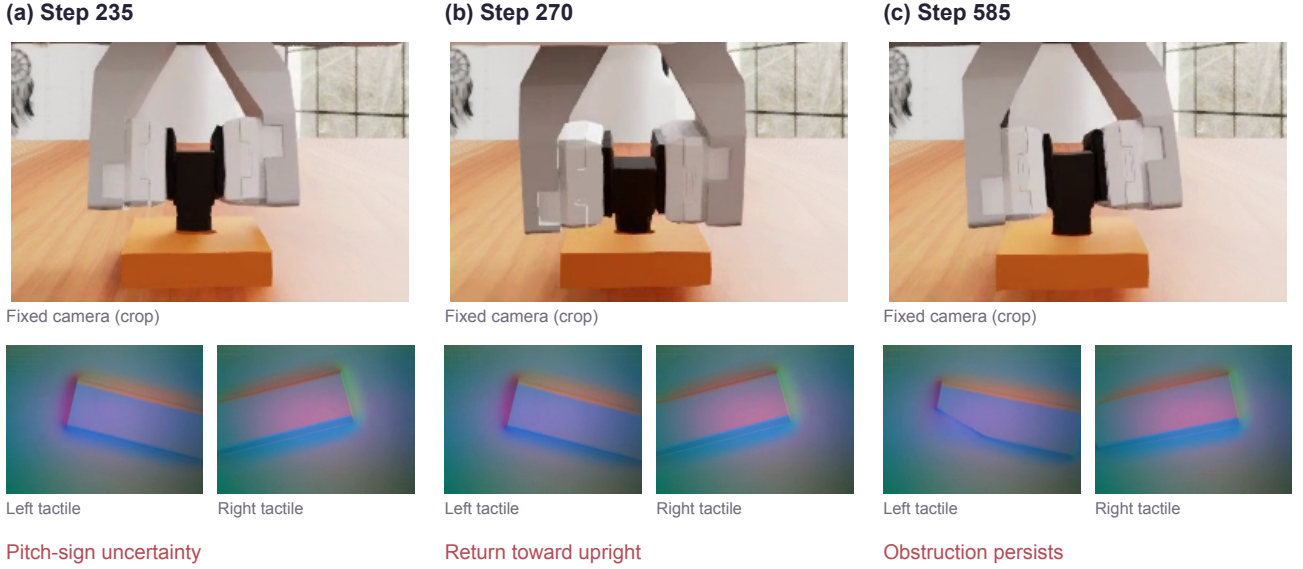


Figure 15. Recognizing obstruction without resolving the correction direction. Insert HDMI, seed 1000062, V+T, failure. Fixed-camera crops and paired tactile images show three stages of the same unsuccessful episode. The recorded notes identify pitch-sign uncertainty at action 235, a return toward an upright interpretation at 270, and persistent obstruction at 585. The changing contact patterns do not lead to successful insertion. [Full rollout](#).

The contact evidence triggers plausible adjustments, but the sequence does not converge on a reliable relationship between the observed deformation and the corrective motion. Depth, pitch, and yaw are revised across attempts while the plug remains near the socket mouth. A new contact pattern can therefore prompt another hypothesis without isolating why the previous attempt failed. Effective diagnosis requires probes that discriminate between competing explanations and subsequent decisions that preserve what those probes established. Together with the Insert Block v1 example, this sequence distinguishes detecting a failed action, identifying its physical cause, and producing an effective correction.

6.3 Active Probing May Fail to Converge

Active probing must reduce uncertainty while preserving enough actions to complete the task. Repeated observations are useful only when they lead to a timely decision about how to proceed.

The Weight Classify comparison in Section 5.3 illustrates how an added modality changes the evidence prioritized by the policy. Comparing contacts under consistent probing conditions is essential for distinguishing a material or load difference from a change in support or grasp geometry.

Empty-Can Selection reveals a related efficiency problem. Repeated measurements can yield a plausible eventual judgment while leaving insufficient opportunity to finish the task. The issue is not only perceptual accuracy, but also whether a probe resolves uncertainty at a reasonable cost in remaining actions.

In the Empty-Can Selection failure (Figure 16), the eventual category judgment and the completed manipulation are separate outcomes. A probe is useful only if the information it produces can still guide the remaining grasp and transfer. Repeatedly changing the contact conditions also makes successive observations harder to compare directly. The correct selection is recorded only at action 370, leaving 30 of 400 actions for the remaining grasp, transport, and placement. This trajectory exposes a stage-transition problem: gathering more evidence must be balanced against the manipulation still required after the classification decision. Consistent probing conditions and a stopping rule should preserve enough opportunity to finish those later stages.

These failures expose a gap between locally plausible actions and reliable task completion. Astra must relate contact cues to the underlying physical constraint, turn that diagnosis into effective corrective motion, and stop probing in time to complete the remaining manipulation.

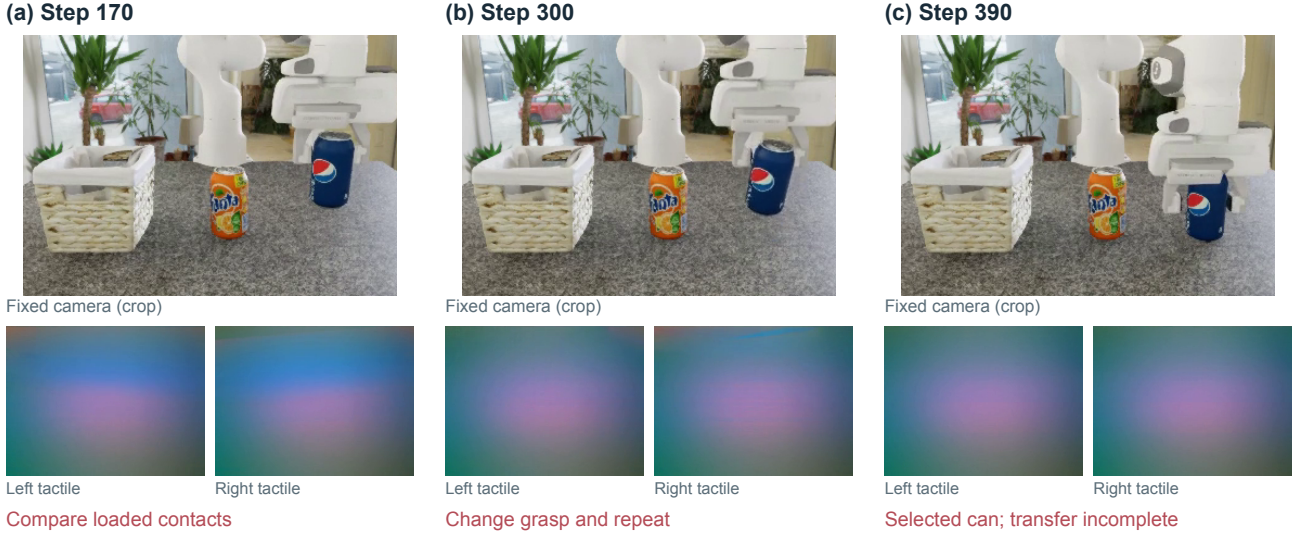


Figure 16. Repeated probing without timely convergence. Empty-Can Selection, seed 10034, V+T, failure. Astra compares load responses, changes the grasp after unstable contact, and repeats the comparison. It selects the correct blue can at step 370 and starts the final grasp-and-lift sequence at step 390, but exhausts the 400-action budget before transfer. Information gathering must be coordinated with a stopping rule and the remaining manipulation. [Full rollout](#).

7 Inference Cost and Latency

The two evaluation phases consume a combined 335.97 million tokens and CNY 7,505.05 in logged provider cost. The V+T phase uses approximately 59% more tokens and incurs 42% more cost than V. Cached inputs account for more than 80% of input tokens in both settings, reflecting the repeated context supplied during closed-loop control.

Table 3. Evaluation-phase token usage and cost. V: vision; V+T: vision and tactile RGB. Phase totals include billed errors, retries, and reruns, but exclude demonstration preparation and later supplementary experiments. Cached input is included in input tokens; reasoning tokens are included in output tokens. Costs sum the provider’s logged estimates.

Input	Input tokens	Cached input	Output tokens	Total tokens	CNY
V	127,250,755	104,837,109	2,283,555	129,534,310	3,102.25
V+T	203,911,092	170,693,217	2,521,218	206,432,310	4,402.80
Total	331,161,847	275,530,326	4,804,773	335,966,620	7,505.05

The median complete-response latency is 15.4 s for V and 14.0 s for V+T; the respective 90th percentiles are 46.3 s and 44.0 s. Both distributions are right-skewed: responses longer than 120 s account for 0.76% and 0.47%, respectively. Latency measures the complete API response for a model request, including provider and gateway waiting time and excluding simulation execution. Each response can generate several control actions. The two modes exhibit similar latency distributions across their respective request mixes and service conditions.

Table 4. Complete API response latency in seconds. V: vision; V+T: vision and tactile RGB. Each row summarizes valid, deduplicated responses with positive output tokens. P90, P95, and P99 denote the 90th, 95th, and 99th percentiles. Mean and quantiles include the full distribution, including responses beyond the displayed histogram range.

Input	Requests	Mean	Median	P90	P95	P99
V	3,927	22.8	15.4	46.3	59.1	110.7
V+T	4,890	21.1	14.0	44.0	56.8	96.6

The response tails summarized in Table 4 delay subsequent decisions in closed-loop operation, since each decision depends on the preceding execution and response. Although each response supplies several

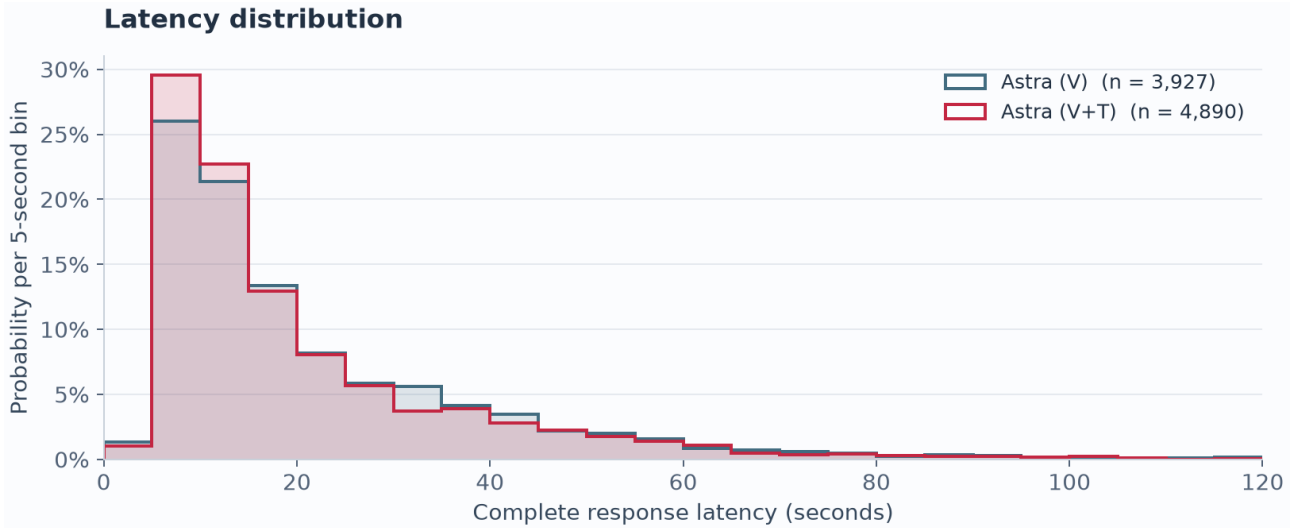


Figure 17. Distribution of API response latency. Five-second bins, normalized by the total number of valid responses in each setting: 3,927 for V and 4,890 for V+T. The plotted range ends at 120 s; the omitted tails remain in the normalization. Failed requests and responses without output tokens are excluded from latency statistics.

control actions, the next observation-dependent correction must wait for another model response. This dependency makes response latency a key factor in end-to-end deployment.

8 Limitations of This Study

All experiments are conducted in simulation; robustness on physical robots remains to be validated under sensor noise and drift, calibration error, actual contact dynamics, and execution delays. Cost constraints limit most tasks to ten episodes per input setting instead of the full 100-episode evaluation, so a single outcome changes the task success rate by ten percentage points. Larger samples, multiple sampling seeds, and repeated executions within the same scenes are needed to quantify variation from scene selection, model decisions, and execution. The experiments also cover only two benchmarks and use task-specific successful demonstrations, with corresponding categories or shapes represented where applicable. Generalization to unfamiliar objects, new tasks, different tactile sensors and robot configurations, and longer multistage manipulation requires systematic evaluation.

9 Conclusion

GPT-6 Astra directly interprets vision-based tactile images and uses their contact cues for manipulation, achieving 93.8% on UniVTAC and 81.8% on OpenViTac and exceeding the reported specialist baselines in our comparison. Tactile images support contact-state inference, cautious grasp formation, alignment, and recovery; fine-grained vision also reveals cues that partially compensate for missing touch. These findings challenge the view that tactile manipulation is intrinsically reserved for specialist policies. The next step is to develop the capability and its evaluation together: joint visual–tactile pretraining can connect contact understanding to foundation-model scaling, while harder tasks with visually hidden physical properties, tighter tolerances, and repeated evaluation on real robots can establish reliable physical competence. The remaining errors in tangential-load interpretation, contact diagnosis, probe selection, and corrective control provide concrete targets for improving policy reliability and the execution harness.

References

- 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>.

- Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. RoboDojo: A Unified Sim-and-Real Benchmark for Comprehensive Evaluation of Generalist Robot Manipulation Policies. *arXiv preprint arXiv:2607.04434*, 2026b. URL <https://arxiv.org/abs/2607.04434>.
- 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. *arXiv preprint arXiv:2609.19138*, 2026. URL <https://arxiv.org/abs/2609.19138>.
- Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://diffusion-policy.cs.columbia.edu/diffusion_policy_2023.pdf.
- Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. AnyTouch: Learning Unified Static-Dynamic Representation across Multiple Visuo-tactile Sensors, 2025. URL <https://arxiv.org/abs/2502.12191>.
- Galbot Team, Xuchuan Chen, Xiaoqian Cheng, Yu Deng, Lihe Ding, Shaocong Dong, Xiangjun Gao, Haozhe Jia, Zekai Li, Zhoujian Li, Yunrui Lian, Sikai Liang, Chenghuai Lin, Dairu Liu, Jiahang Liu, Qingtao Liu, Yuxuan Ma, Zekun Qi, Jiayi Su, He Wang, Ruochen Xu, Tianyu Xu, Xudong Xu, Zhe Xu, Mi Yan, Siming Yan, Li Yi, Ruixi Yu, Jinlu Zhang, Yintianrun Zhang, Zhikai Zhang, Zhizheng Zhang, Yixin Zheng, and Weiyi Zhu. Systematically Exploring the Capabilities of GPT-6 Astra as Embodied Policies. *arXiv preprint arXiv:2609.38537*, 2026. URL <https://arxiv.org/abs/2609.38537>.
- Abraham George, Selam Gano, Pranav Katragadda, and Amir Barati Farimani. ViTaL Pretraining: Visuo-Tactile Pretraining for Tactile and Non-Tactile Manipulation Policies. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 258–264. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128336. URL <https://doi.org/10.1109/ICRA55743.2025.11128336>.
- 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. *arXiv preprint arXiv:2609.22966*, 2026. URL <https://arxiv.org/abs/2609.22966>.
- Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. ViTacFormer: Learning Cross-Modal Representation for Visuo-Tactile Dexterous Manipulation, 2025. URL <https://arxiv.org/abs/2506.15953>.
- 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, 2024. URL <https://arxiv.org/abs/2410.24090>.
- Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning Fine-Grained Manipulation with Visuo-Tactile Sensing, 2024. URL <https://arxiv.org/abs/2410.24091>.
- Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-VLA: Unlocking Vision-Language-Action Model’s Physical Knowledge for Tactile Generalization, 2025. URL <https://arxiv.org/abs/2507.09160>.
- 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. *arXiv preprint arXiv:2307.05973*, 2023. URL <https://arxiv.org/abs/2307.05973>.
- Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL <https://proceedings.mlr.press/v270/kim25c.html>.
- Xiaoqi Li, Muhe Cai, Jiadong Xu, Juan Zhu, Hongwei Fan, Yan Shen, Guangrui Ren, and Hao Dong. AT-VLA: Adaptive Tactile Injection for Enhanced Feedback Reaction in Vision-Language-Action Models, 2026. URL <https://arxiv.org/abs/2605.07308>.
- 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. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023. doi: 10.1109/icra48891.2023.10160591. URL <https://doi.org/10.1109/icra48891.2023.10160591>.
- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*, 2023. URL <https://arxiv.org/abs/2306.03310>.
- Fangcheng Liu, Yeqing Shen, Anda Cheng, Weishi Mi, Chao Tang, Chenyuan Liu, Yushun Xiang, Tingguang Li, Yong-Lu Li, and Yehui Tang. RoboICL: Embodied In-Context Learning with GPT-6 Astra. *arXiv preprint arXiv:2609.34261*, 2026. URL <https://arxiv.org/abs/2609.34261>.

- Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization, 2025. URL <https://arxiv.org/abs/2504.16054>.
- 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/GPT-as-Policy>.
- Kazutoshi Tanaka. TacZero: Training-Free Peg Insertion Using a General-Purpose Vision-Language Model with Tactile Feedback. *arXiv preprint arXiv:2610.07621*, 2026. URL <https://arxiv.org/abs/2610.07621>.
- Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU Parallelized Robot Simulation and Rendering for Generalizable Embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.021. URL <https://www.roboticsproceedings.org/rss21/p021.html>.
- Enyi Wang, Mingxin Wang, Quan Shi, Hetian Guo, Hongyu Wang, Xi Wang, Bin Qian, Yupeng Zheng, Wenxuan Song, Houde Liu, Yong Xu, Cheng Chi, Wenchao Ding, Yilun Chen, and Yan Wang. TacDyn-WAM: Learning Implicit Tactile Dynamics in a Heterogeneous Visuo-Tactile World Action Model. *arXiv preprint arXiv:2610.00638*, 2026. URL <https://arxiv.org/abs/2610.00638>.
- Yifan Wu, Qin Li, Nan Min, Guojin Zhong, Haoyu Zhao, Zhiyuan Li, Houze Xu, Shengqi Xu, Xingyao Lin, Zijie Diao, Zhaoxiang Liu, Shiguo Lian, Shunlin Lu, Shihao Zhao, Ziyi Ye, Zuxuan Wu, and Yu-Gang Jiang. Open-ViTac: Learning and Benchmarking Visuo-Tactile Policies in a Unified Sim-and-Real Framework. *arXiv preprint arXiv:2610.10384*, 2026. URL <https://arxiv.org/abs/2610.10384>.
- Chen Yang, Linzhe Shi, Changjie Wu, Hang Zhang, Ronghan Chen, Lingjun Zhang, Xu Hu, Mu Xu, Jiansheng Fan, and Chen Wang. SimpleTouch: Can Vision-Language-Action Models Master Contact-Rich Manipulation Without Tactile Policy Pretraining? *arXiv preprint arXiv:2610.02784*, 2026a. URL <https://arxiv.org/abs/2610.02784>.
- Zhenjie Yang, Yideng Zhang, Dongjie Zhang, Chenyu Jiang, Xianshuai Liu, Yufeng Li, Zuhao Ge, Xingyu Jiao, Zheng Zhang, Kaiyu He, He Wang, Yuwen Zhong, Yi Deng, Muyun Jiang, Xianliang Huang, Haisheng Su, Donghang Zhang, Jian Zhang, Xue Yang, Hongyang Li, Zuxuan Wu, Yu-Gang Jiang, Xiaosong Jia, and Junchi Yan. Bench2Dex: Benchmarking Visuo-Tactile Bimanual Dexterous Manipulation Across Dexterous Hands. *arXiv preprint arXiv:2609.15726*, 2026b. URL <https://arxiv.org/abs/2609.15726>.
- 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>.
- Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. doi: 10.3390/s17122762. URL <https://doi.org/10.3390/s17122762>.
- Wenbo Zhang, Kaixuan Wang, Yutao Ouyang, Xiaoyu Huang, Liyang Li, Kailun Su, Weiyang Jin, Wenhao Chai, Haotian Liang, Zhiyang Dou, Yue Chen, and Tianxing Chen. An Unexpected Robot Policy: Early Evaluations of GPT-6 Astra on RoboDojo and Beyond. *arXiv preprint arXiv:2609.24170*, 2026. URL <https://arxiv.org/abs/2609.24170>.
- Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward Adelson. Transferable Tactile Transformers for Representation Learning Across Diverse Sensors and Tasks. 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. 3766–3779. PMLR, 06–09 Nov 2025. URL <https://proceedings.mlr.press/v270/zhao25c.html>.
- Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems XIX*, RSS2023. Robotics: Science and Systems Foundation, July 2023. doi: 10.15607/rss.2023.xix.016. URL <https://www.roboticsproceedings.org/rss19/p016.html>.

A Task-Level In-Context Configuration

Table 5 summarizes the archived demonstration configuration. A shot is a distinct demonstration trajectory. Each segment contains H consecutive actions with the associated starting and resulting observations; segments are distributed across the demonstration rather than selected online by the policy. The same trajectories, segment starts, and actions are used in V and V+T. Only V+T additionally includes the two tactile RGB views in demonstrations and online observations.

Table 5. Demonstrations, segment lengths, and camera inputs. “Shots” counts distinct demonstration trajectories; “Segments” is the number selected per demonstration. H is the number of actions per segment. Total actions counts demonstration actions in context, not an evaluation budget. The camera column lists official scene views; the visuo-tactile setting additionally receives both tactile RGB views. Insert Block v2 supplies all three shape demonstrations to each variant.

Task	Shots	Segments	H	Total actions	Cameras
UniVTAC					
Lift Bottle	1	12	10	120	Fixed
Pull-out Key	1	12	5	60	Fixed
Lift Can	1	12	5	60	Fixed + wrist
Put Bottle	1	12	10	120	Fixed
Insert Hole	1	12	5	60	Fixed
Insert HDMI	1	12	5	60	Fixed
Insert Tube	1	12	5	60	Fixed + wrist
Grasp Classify	2	5	4	40	Fixed
OpenViTac					
Weight Classify	2	5	5	50	Fixed + wrist
Hardness Classify	2	5	5	50	Fixed + wrist
Roughness Classify	2	5	5	50	Fixed + wrist
Roughness-Guided Regrasp	2	5	5	50	Fixed + wrist
Empty-Can Selection	2	5	10	100	Fixed + wrist
Grasp Chip	1	12	10	120	Fixed + wrist
Gear Assembly	1	12	10	120	Fixed + wrist
Pull Drawer	1	12	5	60	Fixed + wrist
Insert USB	1	12	5	60	Fixed + wrist
Insert Block v1	1	12	10	120	Fixed + wrist
Insert Block v2	3	5	10	150	Fixed + wrist

Temporal Selection of Demonstration Blocks. For a source trajectory with N observations, each selected start s defines H consecutive actions and the two observations at s and $s + H$. Our preparation calls the released RoboICL deterministic selector (Liu et al., 2026): temporal targets distribute non-overlapping blocks across the trajectory, including its initial and terminal windows, and invalid action windows are excluded. The same selector is used with twelve blocks for one-shot context and five blocks per trajectory for multi-shot context. It selects blocks offline; online actions do not change their starts. For example, the two Weight Classify demonstrations use starts $\{0, 33, 66, 99, 132\}$ and $\{0, 35, 69, 104, 139\}$ with $H = 5$. The corresponding images are the block endpoints, rather than every intermediate action frame.

Execution and Observation Timing. Demonstrations and generated commands use a 60 Hz control cadence, with two 120 Hz simulator substeps per command. Demonstration motion increments and gripper targets are reconstructed from recorded states under the benchmark’s position-control interface. Both input settings use this execution protocol.

B Paired Outcomes on Touch-Improved Tasks

Paired episodes share the evaluation scene and random seed. Panels show each trajectory’s final recorded frame with its own action index. Percentages report task-level success rates. For visual clarity, the figures display one scene-camera view; V+T episodes additionally use paired tactile images during control.

Pull-out Key

Seed 1000025 | 30% -> 90% task success



Vision | step 94 | failed



Vision + touch | step 82 | success

Lift Can

Seed 1000093 | 80% -> 100% task success



Vision | step 69 | failed



Vision + touch | step 92 | success

Insert Hole

Seed 1000014 | 60% -> 100% task success



Vision | step 48 | failed



Vision + touch | step 79 | success

Figure 18. Paired outcomes, part I. Pull-out Key (seed 1000025), Lift Can (1000093), and Insert Hole (1000014). The V episodes fail and the corresponding V+T episodes succeed. These endpoint views complement the successful contact-inference sequences in Figures 4 and 5.

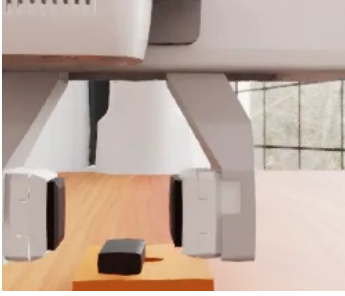
Pull-out Key. The vision-only run repeatedly turns at fixed depth. With touch, growing edge deformation prompts torsional unloading; extraction begins after the contact returns to a round shape. The contact change guides the transition to axial withdrawal.

Lift Can. The vision-only run follows a tipping arc and later reports possible slip. The visuo-tactile run checks bilateral retention before lifting, then rotates the object upright and lowers it without releasing.

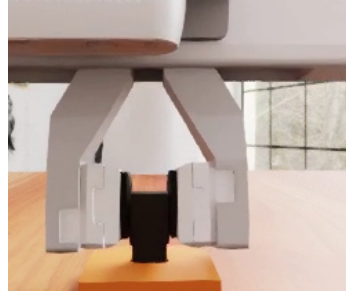
Insert Hole. Vision-only control probes a positive-X slant. Oppositely tilted tactile bands instead guide a negative-X pivot while the tip remains at the entrance, followed by insertion along the inferred bore. Touch helps select the hidden direction before advancement.

Insert HDMI

Seed 1000063 | 50% -> 80% task success



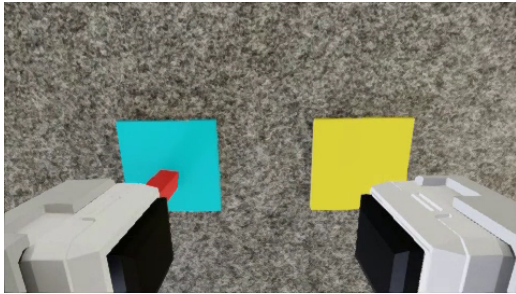
Vision | step 299 | failed



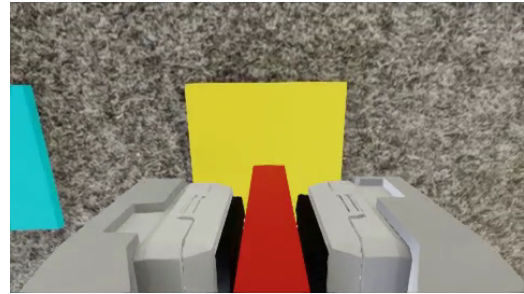
Vision + touch | step 67 | success

Hardness Classify

Seed 10003 | 70% -> 90% task success



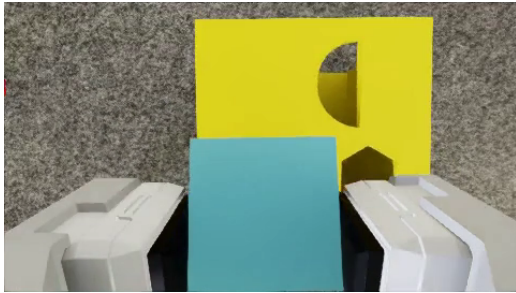
Vision | step 99 | failed



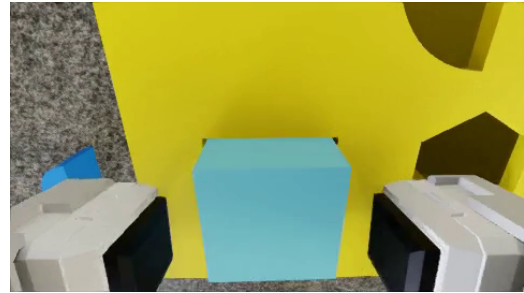
Vision + touch | step 39 | success

Insert Block v1

Seed 10008 | 40% -> 70% task success



Vision | step 199 | failed



Vision + touch | step 109 | success

Figure 19. Paired outcomes, part II. Insert HDMI (seed 1000063), Hardness Classify (10003), and Insert Block v1 (10008). The V+T episodes succeed where the same-seed V episodes fail. Hardness Classify requires the correct category-specific destination; the insertion tasks require the part to enter its matching opening. Complete action sequences and input views remain available in the online archive.

Insert HDMI. The vision-only run ends during another attempt to regrasp the displaced connector. With touch, shifting contact without plug motion is interpreted as rim contact, prompting unloading and fore-aft corrections. Advancement resumes once the plug descends with stable tactile contact. This sequence connects contact interpretation to correction of the object-to-socket alignment.

Hardness Classify. Vision-only control labels the hard block as compliant, places it on the blue plate, and continues observing the stable placement. The visuo-tactile run probes the grasp, associates a sharply bounded indentation with the hard category, and moves to the yellow plate. The pair distinguishes secure object transport from the material judgment that determines the required destination.

Insert Block v1. Both runs reach the square opening. In the vision-only trace, the cube remains on the rim after release, leading to repeated regrasping and centering attempts. The visuo-tactile run maintains retention and checks alignment before a shallow descent and gradual release complete insertion. The outcome depends on maintaining alignment from transport through release.