OPENVITAC: LEARNING AND BENCHMARKING VISUO-TACTILE POLICIES IN A UNIFIED SIM-AND-REAL FRAMEWORK

Anonymous authors
Paper under double-blind review



Figure 1: **Overview of OpenViTac.** OpenViTac establishes a unified benchmark for visuo-tactile manipulation, spanning four complementary capability dimensions: physical-property perception, fragility-aware interaction, contact-rich manipulation, and precision manipulation. It provides task-corresponding simulation and real-world settings with aligned assets, supports multiple gripper configurations and tactile sensors, and enables consistent evaluation of visuo-tactile policies across both domains.

ABSTRACT

Tactile feedback provides embodied agents with physical information beyond visual observations, enabling more reliable interaction with the real world. However, despite the rapid progress of vision-tactile-language-action (VTLA) policies, there remains a lack of unified benchmarks for evaluating tactile-enabled robot manipulation across simulation and the real world. To address this gap, we introduce **OpenViTac**, a visuo-tactile manipulation benchmark for evaluating robot policies across simulation and the real world. OpenViTac organizes contact-rich manipulation into four tactile-relevant capability dimensions and provides paired simulation–real-world settings for consistent evaluation of VLA, WAM, and VTLA policies. Building upon this benchmark, we investigate how different tactile representations and integration strategies affect the performance of pre-

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.

trained VLA models. Correspondingly, we introduce **OpenVTLA**, a tactile augmentation framework that combines the best-performing representation and integration strategy. Furthermore, we leverage the paired benchmark setting to study sim–real co-training and analyze factors affecting cross-domain policy learning. Together, OpenViTac provides a unified platform for evaluating and advancing visuo-tactile robot manipulation.

1 INTRODUCTION

"The ideas of sight and touch make two species entirely distinct and heterogeneous." — George Berkeley

Recent advances in Vision-Language-Action (VLA) models (Brohan et al., 2023; Kim et al., 2025) have enabled robots to leverage large-scale visual and linguistic knowledge for increasingly generalizable manipulation (Intelligence et al., 2025). These foundation policies have demonstrated strong performance across diverse manipulation scenarios (Liu et al., 2023; Khazatsky et al., 2024), yet remain primarily grounded in visual observations. However, many real-world interactions require understanding physical states that are not directly observable from vision alone, such as object properties, contact dynamics, and interaction forces (Feng et al., 2026). By providing direct feedback from physical interaction, tactile sensing (Dong et al., 2017; Team et al., 2026b) augments visual perception with complementary physical information, enabling embodied agents to better understand and interact with the physical world, such as contact geometry, force cues, and material-dependent responses. This has motivated a shift from purely vision-based policies toward Vision-Tactile-Language-Action (VTLA) models (Huang et al., 2025; Yuan et al., 2026a; Niu et al., 2026), which combine global scene understanding from vision with local interaction feedback from touch.

Driven by the promise of tactile feedback, recent VTLA studies have rapidly expanded the design space of tactile-enabled robot policies, exploring different tactile representations (Zhao et al., 2025; Higuera et al., 2025; Feng et al., 2026), fusion strategies (Morissette et al., 2026; Xu et al., 2026), and policy architectures (Yuan et al., 2026a; Niu et al., 2026). Nevertheless, these approaches are typically evaluated under different sensors, tasks, robot embodiments, and experimental protocols, making systematic comparison challenging and leaving the contribution of tactile sensing insufficiently understood. Existing manipulation benchmarks, such as LIBERO (Liu et al., 2023), RoboTwin (Chen et al., 2025), and RoboDojo (Chen et al., 2026b), primarily focus on vision-based policies and do not support tactile sensing. Recent visuo-tactile benchmarks, such as Tabero (Wu et al., 2026a) and UniVTAC (Chen et al., 2026a), have started to introduce tactile observations and contact-rich tasks for evaluation. However, as shown in Tab. 1, existing benchmarks still lack comprehensive coverage of tactile-relevant capabilities, diverse hardware configurations, and sim–real correspondence.

To address this, we introduce **OpenViTac**, a unified benchmark for evaluating visuo-tactile robot manipulation across simulation and the real world. As shown in Fig. 1, OpenViTac organizes tactile-enabled manipulation into four capability dimensions: physical property perception, fragility-aware manipulation, contact-rich manipulation, and precision manipulation, covering diverse scenarios that require understanding object properties, contact states, and fine-grained interaction dynamics. We evaluate representative VLA, WAM, and VTLA policies (Intelligence et al., 2025; Community, 2026; Zheng et al., 2026; Wu et al., 2026b; Ma et al., 2026; Team et al., 2026d; Ali et al., 2025; Yuan et al., 2026b; Li et al., 2026; Team et al., 2026c; Yuan et al., 2026a; Team et al., 2026a) on OpenViTac, providing a comprehensive comparison of their tactile-enabled manipulation capabilities.

Building upon OpenViTac, we further investigate how tactile observations can be incorporated into vision-based policies through two design choices: tactile representation and tactile integration. Our study reveals that stronger tactile representations and effective integration strategies consistently improve performance on contact-sensitive manipulation tasks. Based on this analysis, we introduce **OpenVTLA**, a tactile augmentation framework for pretrained VLA models. Quantitatively, the strongest VLA, WAM, and VTLA baselines achieve 51.5%, 53.3%, and 61.2% average success rates in simulation, respectively, while OpenVTLA reaches 68.7%. The same trend holds in real-world evaluation, where OpenVTLA achieves 54.6%, compared with 43.1%, 48.5%, and 51.9% from the strongest VLA, WAM, and VTLA baselines. Finally, leveraging the paired sim–real setting

Table 1: **Comparison of representative simulated manipulation benchmarks with OpenViTac.** We include recent visuo-tactile benchmarks together with representative vision-based benchmarks.

Benchmark	Gripper Types	Tactile Sensing	Tactile Sensor Types	Property Perception	Fragility-/Force-Aware	Precision Manipulation	Scene Diversity
CALVIN (Mees et al., 2022)	1	✓	1	✗	✗	✗	✗
LIBERO (Liu et al., 2023)	1	✗	-	✗	✗	✗	✗
ManiFeel (Luu et al., 2025)	1	✓	1	✗	✗	✗	✗
Tabero (Wu et al., 2026a)	1	✓	1	✗	✓	✗	✗
UniVTAC (Chen et al., 2026a)	1	✓	1	✓	✗	✓	✗
NeoSim (Team et al., 2026b)	1	✓	1	✗	✓	✓	✗
SoftVTBench (Jing et al., 2026)	1	✓	1	✗	✓	✗	✗
OpenViTac (Ours)	2	✓	3	✓	✓	✓	✓

of OpenViTac, we investigate sim–real co-training as a strategy for improving real-world policy learning through additional simulation data.

Overall, our contributions are threefold:

- **A unified visuo-tactile manipulation benchmark.** We introduce **OpenViTac**, a tactile-centric benchmark with four capability dimensions and paired simulation–real-world settings for evaluating VLA, WAM, and VTLA policies.
- **An investigation of tactile adaptation for pretrained VLAs.** We investigate tactile representation and integration strategies for adapting vision-based policies to touch, leading to **OpenVTLA**, a tactile augmentation framework achieving strong performance across simulation and real-world evaluations.
- **A sim–real co-training study.** We leverage OpenViTac's paired sim–real setting to investigate sim–real co-training and analyze the factors affecting cross-domain policy learning.

2 Related Work

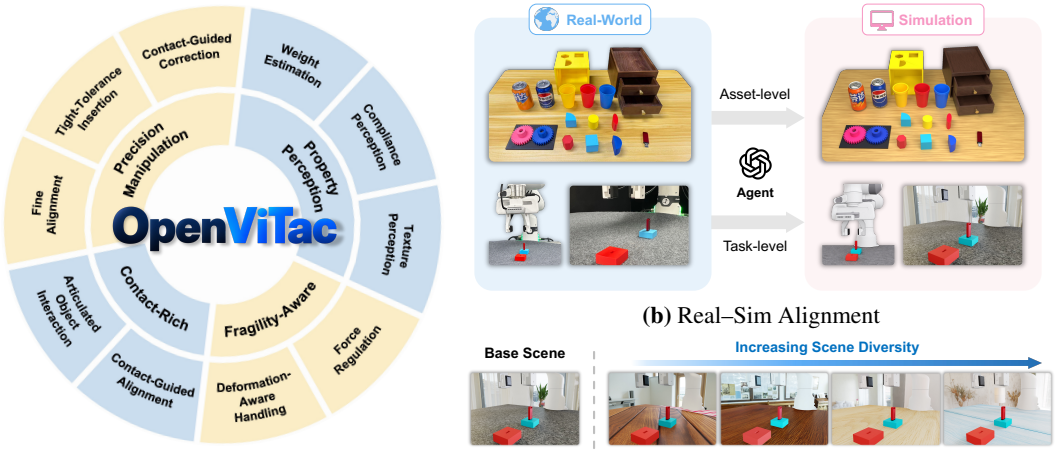
Here we review related work on visuo-tactile robot learning and visuo-tactile datasets and benchmarks. Additional discussion can be found in Appendix A.

2.1 Visuo-Tactile Robot Learning

Recent robot foundation policies have largely developed around vision-language-action (VLA) models and world-action models (WAMs) (Kim et al., 2025; Intelligence et al., 2025). Tactile feedback provides complementary contact and physical-property information that can improve contact-sensitive manipulation, motivating tactile-augmented policies such as VTLA models (Yuan et al., 2026a; Team et al., 2026a; Niu et al., 2026; Wang et al., 2026) and tactile-aware world-action models (Lou et al., 2026; Team et al., 2026c). Existing approaches, however, differ substantially in how touch is incorporated, including feature modulation (Morissette et al., 2026), action-level conditioning (Team et al., 2026a), dedicated tactile experts (Yuan et al., 2026a), and late-stage action refinement (Niu et al., 2026; Wang et al., 2026). Given the lack of consensus on how tactile information should be incorporated into pretrained VLAs, in this paper we systematically study representative tactile integration strategies under a shared $\pi_{0.5}$ backbone.

2.2 Visuo-Tactile Datasets and Benchmarks

General manipulation benchmarks such as LIBERO (Liu et al., 2023), RoboTwin (Chen et al., 2025), and RoboDojo (Chen et al., 2026b) provide standardized evaluation across diverse simulated tasks, but primarily focus on vision-based manipulation. A broad range of tactile datasets has been developed for representation learning (Yang et al., 2022; Kerr et al., 2022; Higuera et al., 2025; Feng et al., 2026), while tactile-enabled real-robot manipulation data have only recently begun to scale across tasks, sensors, and embodiments (Yuan et al., 2026a; Team et al., 2026b; Lou et al., 2026). For policy-level evaluation, ManiFeel (Luu et al., 2025) and TacO (Zorin et al., 2026) study tactile sensing in downstream manipulation, while UniVTAC (Chen et al., 2026a) and NeoSim (Team



(a) Capability Taxonomy

(b) Real–Sim Alignment

(c) Scene Diversity

Figure 2: **Overview of the OpenViTac design.** (a) Capability taxonomy with four visuo-tactile capabilities and their underlying skills. (b) Construction of task-corresponding simulation environments from real-world references at the asset- and task-levels. (c) Support for scene diversification through configurable appearance variations in simulation.

et al., 2026b) further provide systematic evaluation in simulation. However, existing simulation benchmarks still exhibit substantial discrepancies from real-world settings. OpenViTac provides a unified tactile-centric benchmark with aligned sim–real tasks and additional real-world data across multiple tactile sensors.

3 OpenViTac

We introduce **OpenViTac**, a unified benchmark for evaluating visuo-tactile robot manipulation across simulation and the real world. OpenViTac organizes tactile-enabled manipulation into four capability dimensions and provides 11 simulation tasks with 8 paired real-world tasks for systematic evaluation of VLA, WAM, and VTLA policies. Building upon OpenViTac, we study tactile adaptation for pretrained VLA models through tactile representation and integration strategies, leading to **OpenVTLA**, and further investigate sim–real co-training enabled by the paired benchmark setting.

3.1 Benchmark

Benchmark design. OpenViTac evaluates whether policies can exploit tactile feedback when visual observations provide incomplete information about physical interaction. As illustrated in Fig. 2a, we organize the benchmark around four visuo-tactile capabilities: *property perception*, *fragility-aware*, *contact-rich*, and *precision*. Further details are provided in Appendix C.

Task suite. We organize the tasks into four complementary visuo-tactile capabilities: (1) *Property Perception* (PP) evaluates whether tactile feedback can reveal visually ambiguous physical properties and support property-conditioned decisions. It includes *weight classification* (WC), *hardness classification* (HC), and *roughness classification* (RC) for property-conditioned placement; *roughness-guided regrasp* (RR) for property-aware regrasping; and *empty-can selection* (ECS) for distinguishing empty from filled cans through physical interaction. (2) *Fragility-Aware* (FA) evaluates force-sensitive manipulation of delicate objects. *Grasp chip* (GC) requires grasping and placing a fragile chip without damaging it. (3) *Contact-Rich* (CR) evaluates manipulation that requires sustained and effective physical contact. It includes *gear assembly* (GA), which requires maintaining gear engagement during assembly, and *pull drawer* (PD), which requires maintaining a stable grasp while actuating an articulated drawer. (4) *Precision Manipulation* (PR) evaluates contact-guided manipulation under tight geometric tolerances. It includes *insert USB* (In-USB) and *insert block v1* (In-B v1) for precise alignment and insertion, and *insert block v2* (In-B v2), which additionally requires selecting the instructed object before insertion.

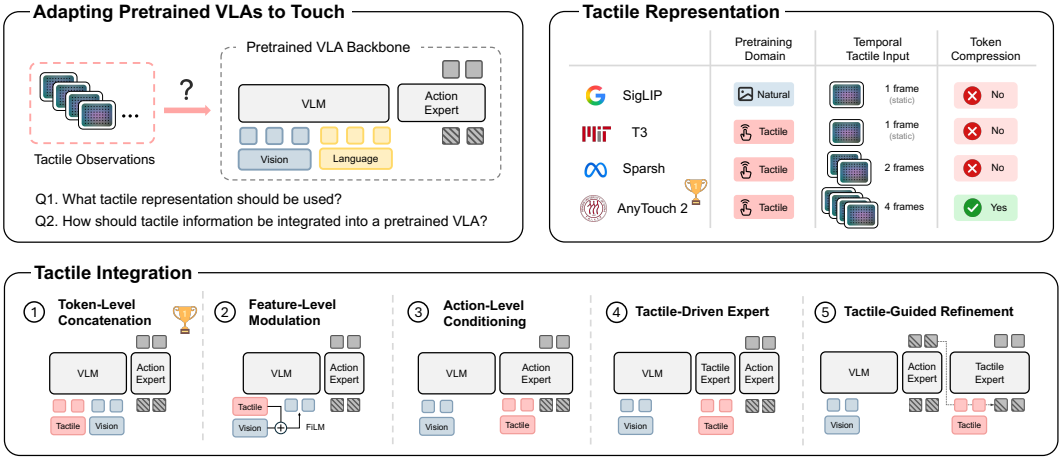


Figure 3: **Design space for adapting pretrained VLAs to touch.** We investigate two complementary dimensions: *tactile representation* and *tactile integration*.

Real–sim alignment. As shown in Fig. 2b, we provide an agent-assisted real-to-sim pipeline for constructing task-corresponding simulation environments from real-world references. (1) At the asset level, multi-view images of real-world objects are used to reconstruct their simulation counterparts, with the agent analyzing task-relevant structure and decomposing objects into functional parts when necessary; (2) At the task level, reference object and target configurations together with robot initial states are used to instantiate the corresponding simulation task and construct executable sub-actions for trajectory collection. OpenViTac further supports scene diversification by varying backgrounds, workspace appearance, and surrounding context while preserving the underlying task configuration, as illustrated in Fig. 2c, providing a flexible way to broaden the simulated data distribution for downstream training.

Evaluation protocol. We evaluate all policies under predefined and balanced task conditions using task-specific binary success criteria. Further details are provided in Appendix C.

3.2 ADAPTING PRETRAINED VLAS TO TOUCH

Large-scale visuo-tactile robot data remain limited and heterogeneous across sensors, embodiments, and manipulation tasks, making it difficult to train tactile-native policies at the scale of modern VLAs (Yuan et al., 2026a; Team et al., 2026a). We therefore study a practical post-training setting in which all variants start from the same pretrained $\pi_{0.5}$ (Intelligence et al., 2025) backbone, while varying how tactile observations are represented and incorporated into the policy. As shown in Fig. 3, we investigate these choices along two dimensions: *tactile representation* and *tactile integration*.

Tactile Representation. The first design choice is how tactile observations should be represented for a pretrained VLA. The most direct option is to reuse SigLIP (Zhai et al., 2023), the native visual encoder of $\pi_{0.5}$, which preserves the existing representation interface but is pretrained on natural images rather than tactile observations. We therefore further consider recent tactile-specific representations, including T3 (Zhao et al., 2025), Sparsh (Higuera et al., 2025), and AnyTouch2 (Feng et al., 2026). T3 processes individual tactile frames, while Sparsh and AnyTouch2 incorporate short temporal histories to capture evolving contact information. AnyTouch2 further adopts a compact tactile representation, consistent with the more localized and contact-specific information carried by tactile observations. We compare these representations under the same token-level concatenation interface in Sec. 4.3; additional background is provided in Appendix A.1.

Tactile Integration. Given a tactile representation, the second design choice is where and how tactile information enters the pretrained VLA. As illustrated in Fig. 3, we consider five representative strategies:

① **Token-Level Concatenation.** Tactile tokens are concatenated with the vision–language sequence, allowing tactile information to participate in VLM processing from the input stage.

② **Feature-Level Modulation.** Tactile features modulate intermediate visual representations through FiLM (Perez et al., 2018), preserving the pretrained VLA pathway as much as possible (Morissette et al., 2026).

③ **Action-Level Conditioning.** Tactile features bypass the VLM and directly condition the action expert, leaving the vision–language backbone unchanged, as in N0-VTLA (Team et al., 2026a).

④ **Tactile-Driven Expert.** A dedicated tactile expert provides a separate computation pathway for tactile representations alongside the action expert, as in FTP-1 (Yuan et al., 2026a).

⑤ **Tactile-Guided Refinement.** Tactile information is introduced at later action-generation stages to refine partially generated actions rather than conditioning the full generation process, as in T-Rex (Niu et al., 2026).

OpenVTLA. Based on the above study, we instantiate OpenVTLA with AnyTouch2 and ① token-level concatenation. AnyTouch2 encodes the tactile history into compact tokens, which are projected to the VLM hidden dimension and prepended to the visual token sequence. This design introduces no additional fusion module or tactile-specific expert, preserving the original VLM and action-generation pathway of $\pi_{0.5}$.

3.3 Simulation–Real-World Co-training

We further study sim–real co-training on OpenViTac, asking whether simulated demonstrations can improve the real-world performance of OpenVTLA under a fixed real-data budget. For each task, we keep the real demonstrations $\mathcal{D}_r$ fixed and augment them with successful simulated demonstrations $\mathcal{D}_s$ collected across the diversified scene configurations in Fig. 2c.

Sim-real data correspondence. Following the task-aware digital-cousin principle (Maddukuri et al., 2025), we preserve the robot embodiment, task objectives, language instructions, object categories, and semantic success criteria across domains. We align action and proprioceptive representations, coordinate frames, units, gripper conventions, and control frequency, while approximately matching the real main and wrist camera viewpoints. Within this shared task structure, simulated scene diversity broadens the training distribution beyond the available real demonstrations.

Mixed-domain training. We retain the OpenVTLA architecture and jointly train on real and simulated demonstrations with

$$\mathcal{L}_{\text{co}} = (1-\alpha)\mathcal{L}_{\text{BC}}(\mathcal{D}_r) + \alpha\mathcal{L}_{\text{BC}}(\mathcal{D}_s), \tag{1}$$

where $\mathcal{L}_{\text{BC}}$ denotes the flow-matching behavior-cloning loss and α specifies the simulation sampling probability independently of dataset size. Training requires neither paired cross-domain trajectories nor an explicit feature-alignment loss. Real-only and co-trained policies share the same real demonstrations, architecture, pretrained initialization, and optimization budget. We also apply this protocol to vision-only $\pi_{0.5}$ to examine how tactile input affects the benefit of simulated data.

4 Experiments

4.1 Experimental Setup

We benchmark representative policies from three model families on OpenViTac, including six VLA models, four WAM models, and two existing VTLA models, together with our OpenVTLA. The VLA baselines include $\pi_{0.5}$ (Intelligence et al., 2025), StarVLA (Qwen3) (Community, 2026), InternVLA-A1.5 (Ma et al., 2026), XVLA (Zheng et al., 2026), Lingbot-VLA-2 (Wu et al., 2026b), and Xiaomi-Robotics (Team et al., 2026d). For WAMs, we evaluate FastWAM (Yuan et al., 2026b), Lingbot-VA (Li et al., 2026), StarVLA (Cosmos) (Ali et al., 2025), and N0-TWAM (Team et al., 2026c). For VTLAs, we compare against FTP-1 (Yuan et al., 2026a) and N0-VTLA (Team et al., 2026a).

All policies are trained and evaluated in a single-task setting, with a separate policy instantiated for each benchmark task. Evaluation follows predefined task conditions in both simulation and the real world; for tasks with multiple variants, each variant is evaluated separately, and the results are aggregated at the task level. The reported Avg. is the unweighted mean of task-level success rates. Implementation and training details for all evaluated policies are provided in Appendix D.

Table 2: Comparison of VLA, WAM, and VTLA models on OpenViTac **in the simulation environment**. **Bold** denotes the best results and underline denotes the second-best.

Models		PP					FA	CR		PR			Avg.
		WC	HC	RC	RR	ECS	GC	GA	PD	In-USB	In-B v1	In-B v2	
VLAs	$\pi_{0.5}$	53	60	64	52	48	39	87	99	14	9	**9.3**	48.6
	StarVLA (Qwen3)	44	38	42	35	32	10	26	35	4	0	0	24.2
	InternVLA-A1.5	53	76	65	77	47	53	79	98	3	6	**9.3**	51.5
	XVLA	28	30	56	47	45	42	84	76	2	3	0.7	37.6
	Lingbot-VLA-2	52	58	58	53	52	41	78	92	8	5	2.7	45.4
	Xiaomi-Robotics	52	46	54	62	38	24	73	84	5	2	0	40.0
WAMs	FastWAM	49	84	67	7	9	50	51	88	6	0	0	37.4
	Lingbot-VA	50	44	39	40	44	22	59	15	2	2	6.0	29.4
	StarVLA (Cosmos)	51	67	94	9	13	1	33	22	3	1	3.3	27.0
	N0-TWAM	85	52	**100**	51	41	57	73	**100**	8	**12**	7.0	53.3
VTLAs	FTP-1	61	**100**	**100**	78	50	60	83	94	**27**	**12**	8.3	61.2
	N0-VTLA	52	97	**100**	55	59	44	51	87	17	0	0	51.1
	OpenVTLA (Ours)	**89**	99	**100**	**90**	**73**	**69**	**91**	99	**27**	11	8.0	**68.7**

Table 3: Comparison of VLA, WAM, and VTLA models on OpenViTac **in the real-world environment**. **Bold** denotes the best results and underline denotes the second-best.

Models		PP		FA	CR		PR			Avg.
		RC	ECS	GC	GA	PD	In-USB	In-B v1	In-B v2	
VLAs	$\pi_{0.5}$	10/20	10/20	16/20	7/20	12/20	3/20	**5/20**	4/30	41.0
	InternVLA-A1.5	10/20	10/20	17/20	9/20	13/20	3/20	3/20	**6/30**	43.1
	Lingbot-VLA-2	10/20	10/20	15/20	3/20	10/20	1/20	2/20	5/30	34.0
	Xiaomi-Robotics	10/20	10/20	16/20	5/20	8/20	1/20	3/20	2/30	34.0
WAMs	FastWAM	8/20	9/20	12/20	5/20	10/20	0/20	1/20	2/30	29.0
	N0-TWAM	18/20	16/20	16/20	8/20	12/20	3/20	2/20	4/30	48.5
VTLAs	FTP-1	18/20	14/20	**19/20**	9/20	**14/20**	3/20	4/20	3/30	51.9
	N0-VTLA	16/20	15/20	17/20	**11/20**	12/20	2/20	3/20	5/30	49.6
	OpenVTLA (Ours)	**19/20**	**17/20**	**19/20**	10/20	12/20	**4/20**	3/20	5/30	**54.6**

4.2 Main Results

Evaluation in simulation. The results of simulation in Tab. 2 reveal clear performance differences across policy families. Vision-only WAMs generally underperform stronger VLA baselines, while N0-TWAM substantially improves over previous WAMs and achieves an average success rate of 53.3%, exceeding all evaluated VLAs. Overall, tactile-enabled policies achieve the strongest performance, suggesting that tactile feedback provides complementary information beyond visual observations for contact-sensitive manipulation. The benefits of tactile sensing are particularly evident on *Property Perception* (PP) and *Fragility-Aware* (FA) tasks, where successful execution requires understanding physical properties and regulating contact interactions. Meanwhile, *Contact-Rich* (CR) tasks remain competitive for strong VLAs, indicating that visual planning and reliable action execution remain important even with tactile feedback. In contrast, *Precision Manipulation* (PR) remains challenging across all model families, highlighting the difficulty of achieving fine-grained contact control. Overall, OpenVTLA achieves the highest benchmark average of 68.7%, outperforming FTP-1 and the strongest VLA baseline by 7.5 and 17.2 percentage points, respectively.

Evaluation in the real world. The real-world results in Tab. 3 exhibit a consistent trend with simulation. VLA baselines achieve average success rates ranging from 34.0% to 43.1%, while the vision-only FastWAM obtains 29.0%. In comparison, the tactile-enabled N0-TWAM achieves 48.5%, and VTLA policies further improve performance to 49.6–54.6%. Across capability dimensions, tactile-enabled policies consistently demonstrate advantages on tasks involving physical state understanding and contact regulation. For example, PP tasks benefit substantially from tactile feedback, with OpenVTLA improving performance on RC and ECS tasks, while CR and FA tasks also maintain competitive performance compared with vision-only policies. In contrast, precision manipulation remains challenging across all policy families, suggesting that fine-grained contact control

requires further improvements in both perception and policy learning. Overall, OpenVTLA achieves the highest real-world average success rate of 54.6%.

Sim–real consistency. Following SIMPLER (Li et al., 2024), we assess whether simulation preserves relative policy performance rather than requiring identical success rates across domains. Across nine methods with complete results on the eight shared tasks, the task-averaged scores yield Pearson $r = 0.913$ and MMRV $= 0.027$, indicating strong agreement between simulation and real-world evaluation. OpenVTLA and FTP-1 rank first and second in both domains. These results support OpenViTac as a useful simulation proxy for comparative policy evaluation.

Tactile benefits depend on how touch is incorporated. On the eight tasks shared across simulation and the real world, OpenVTLA improves over $\pi_{0.5}$ by 13.6 and 13.5 percentage points, respectively. At the same time, the substantial variation among tactile-enabled policies—from N0-VTLA and N0-TWAM to FTP-1 and OpenVTLA—shows that access to tactile observations alone does not guarantee consistent gains. This motivates the controlled study in Sec. 4.3, where we examine how tactile representation and integration determine the effectiveness of adapting pretrained VLAs to touch.

4.3 Understanding Tactile Adaptation in Pretrained VLAs

What tactile representation is effective? We first compare different tactile representations under the same token-level concatenation interface. As shown in Fig. 4, the choice of tactile representation has a substantial impact on downstream performance. After normalizing task-level results to percentages, AnyTouch2 achieves the highest average score of 68.7%, outperforming Sparsh (64.7%), SigLIP (59.5%), and T3 (58.1%). The advantage is particularly pronounced on tasks that require reasoning about evolving contact states; for example, AnyTouch2 improves over Sparsh by 12 percentage points on ECS and 10 points on USB insertion. More generally, the temporally-aware representations, Sparsh and AnyTouch2, tend to outperform the frame-wise alternatives, suggesting that tactile dynamics provide useful information beyond static contact appearance. We further examine two factors underlying this trend: (a) replacing the pretrained AnyTouch2 encoder with a randomly initialized counterpart leads to a clear performance degradation, indicating that tactile-specific pretraining provides useful representations for VLA adaptation; (b) removing temporal variation by replacing the tactile history with repeated copies of the current frame particularly degrades physical-property recognition, including hardness and weight discrimination. These controlled studies support the importance of both tactile pretraining and temporal dynamics; detailed results are provided in Appendix E.

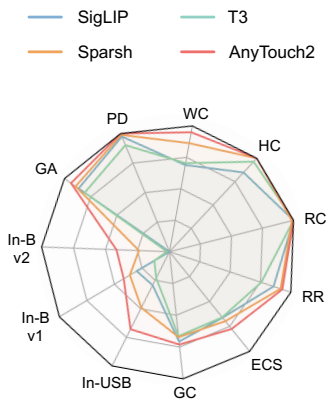


Figure 4: Performance of different tactile representations under the same token-level concatenation strategy.

Where should tactile information be integrated? We next compare representative tactile integration strategies in Fig. 3, with results summarized across PP, FA, CR, and PR in Tab. 4. Overall, ① token-level concatenation achieves the highest average performance and leads on PP, FA, and CR, while remaining competitive on PR. Among the AnyTouch2-based variants, introducing tactile information directly at the VLM input substantially outperforms ③ action-level conditioning and ⑤ late refinement, suggesting that early access to touch is beneficial for learning contact-dependent representations and decisions. A possible explanation is that token-level concatenation allows tactile information

Table 4: **Tactile integration comparison.** Scores are averaged over four task groups. $\pi_{0.5}{}^{\dagger}$ denotes variants obtained by adapting the pretrained $\pi_{0.5}$ backbone with different tactile integration strategies. **Bold** denotes the best results and underline denotes the second-best.

Method	Representation	Integration	PP	FA	CR	PR	Avg.
OpenVTLA	AnyTouch2	① Concat	**90.2**	**69.0**	**95.0**	15.3	**68.7**
$\pi_{0.5}{}^{\dagger}$	Sparsh	② Modulation	82.2	49.0	73.0	5.0	56.5
$\pi_{0.5}{}^{\dagger}$	AnyTouch2	③ Action	70.8	43.0	62.0	4.7	48.6
N0-VTLA	DINOv2	③ Action	72.6	44.0	69.0	5.7	51.1
FTP-1	T3	④ Expert	77.8	60.0	88.5	**15.8**	61.2
$\pi_{0.5}{}^{\dagger}$	AnyTouch2	⑤ Refine	63.0	51.0	81.0	8.7	50.4

to interact with the full pretrained VLM throughout policy computation, rather than restricting it to downstream action-generation stages. Together with the representation study above, these results support a simple recipe for tactile adaptation of pretrained VLAs: use a pretrained temporally-aware tactile encoder and inject its tokens directly into the VLM input sequence.

4.4 Simulation–Real-World Co-training

We study whether simulated demonstrations can improve real-world performance under a fixed real-data budget on Insert USB (In-USB) and Gear Assembly (GA), using both vision-only $\pi_{0.5}$ and OpenVTLA. For each task, all policies are trained with 100 real demonstrations, with co-training additionally incorporating either 250 or 500 simulated demonstrations collected across diversified scene configurations. We further evaluate the real-only and 500-simulation policies under lighting perturbations. Each model–task–condition setting is evaluated over 20 trials.

Diverse simulated data improves real-world performance. As shown in Tab. 5, adding simulated demonstrations improves both policies without increasing the real-data budget. With 500 simulated demonstrations, $\pi_{0.5}$ improves from 15% to 40% on In-USB and from 35% to 50% on GA. OpenVTLA improves from 20% to 40% and from 50% to 60%, respectively. Averaged over the two tasks, these correspond to gains of 20 percentage points for $\pi_{0.5}$ and 15 points for OpenVTLA. The smaller gain for OpenVTLA may partly reflect residual differences in contact-induced tactile signals across domains, motivating consistent sensor calibration and improved tactile simulation.

Table 5: **Sim–real co-training and robustness.** All settings use 100 real demonstrations per task; "Sim." denotes the number of additional simulated demonstrations.

Model	Sim.	Std. Lighting		Lighting Perturb.	
		In-USB	GA	In-USB	GA
$\pi_{0.5}$	0	3/20	7/20	0/20	2/20
	250	5/20	9/20	–	–
	500	8/20	10/20	4/20	8/20
OpenVTLA	0	4/20	10/20	2/20	10/20
	250	6/20	10/20	–	–
	500	8/20	12/20	6/20	12/20

Co-training and touch improve performance under lighting perturbations. Under perturbed lighting, real-only $\pi_{0.5}$ falls to 0% on In-USB and 10% on GA. Adding 500 simulated demonstrations raises these results to 20% and 40%, showing that broader simulated training supports better performance under visual disturbance, although degradation relative to standard lighting remains. OpenVTLA is less sensitive to the perturbation: its real-only policy achieves 10% on In-USB and retains 50% on GA, while co-training yields 30% and 60%. In particular, its GA success rate is unchanged by lighting perturbations under both training settings. This pattern is consistent with tactile feedback providing contact information that is less sensitive to ambient lighting, complementing the robustness gained from diverse simulated experience.

5 Conclusion

In this work, we introduce **OpenViTac**, a visuo-tactile manipulation benchmark that enables systematic evaluation of tactile-enabled robot policies across simulation and the real world. By organizing manipulation scenarios into four tactile-relevant capability dimensions, OpenViTac provides a unified testbed for comparing VLA, WAM, and VTLA policies under consistent settings. Building upon this benchmark, we further investigate how tactile observations can be incorporated into pretrained VLA models through different representation and integration strategies, leading to **OpenVTLA**, a tactile augmentation framework that achieves strong performance across both simulation and real-world evaluations. We further explore sim–real co-training enabled by the paired benchmark setting, demonstrating the potential of leveraging simulated experience to improve real-world manipulation. Despite these advances, several challenges remain for future visuo-tactile manipulation benchmarks. In particular, the correspondence between simulated and real tactile observations is still limited by the fidelity of tactile rendering, sensor modeling, and multimodal calibration. Future work could further improve simulation realism through more accurate sensor and camera calibration, while expanding benchmark coverage toward more complex tactile-centric interactions, such as deformable object manipulation, surface interaction, and long-horizon household tasks.

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229, pp. 2165–2183. PMLR, 2023.

Baijun Chen, Weijie Wan, Tianxing Chen, Xianda Guo, Congsheng Xu, Yuanyang Qi, Haojie Zhang, Longyan Wu, Tianling Xu, Zixuan Li, et al. Univtac: A unified simulation platform for visuo-tactile manipulation data generation, learning, and benchmarking. *arXiv preprint arXiv:2602.10093*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026b.

Yizhou Chen, Andrea Sipos, Mark Van der Merwe, and Nima Fazeli. Visuo-tactile transformers for manipulation. *arXiv preprint arXiv:2210.00121*, 2022.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Siyuan Dong, Wenzhen Yuan, and Edward H Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 137–144. IEEE, 2017.

Ruoxuan Feng, Yuxuan Zhou, Siyu Mei, Dongzhan Zhou, Pengwei Wang, Shaowei Cui, Bin Fang, Guocai Yao, and Di Hu. Anytouch 2: General optical tactile representation learning for dynamic tactile perception. *arXiv preprint arXiv:2602.09617*, 2026.

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. 270: 885–915, 06–09 Nov 2025.

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. *arXiv preprint arXiv:2507.09160*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. \pi_ {0.5}: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Bowen Jing, Mingxin Wang, Ruiyang Hao, Chenchen Ge, Hanwen Shen, Junjie He, Yang Cui, Yiming Hou, Weitao Zhou, Jiawei Wang, et al. Softvtbench: A deformation-aware visuo-tactile dataset and benchmark for deformable-object manipulation. *arXiv preprint arXiv:2608.18701*, 2026.

Justin Kerr, Huang Huang, Albert Wilcox, Ryan Hoque, Jeffrey Ichnowski, Roberto Calandra, and Ken Goldberg. Self-supervised visuo-tactile pretraining to locate and follow garment features. *arXiv preprint arXiv:2209.13042*, 2022.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems (RSS)*, Delft, Netherlands, 2024. doi: 10.15607/RSS.2024.XX.120.

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. 270:2679–2713, 06–09 Nov 2025.

Yu Lei, Minghuan Liu, Abhiram Maddukuri, Zhenyu Jiang, and Yuke Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies. *arXiv preprint arXiv:2604.13645*, 2026.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Yunfan Lou, Yifan Ye, Yankai Fu, Jun Cen, Xiaowei Chi, Yaoxu Lyu, Peidong Jia, Sirui Han, Zhihe Lu, and Shanghang Zhang. Dream-tac: A unified tactile world action model for contact-rich robot manipulation. *arXiv preprint arXiv:2606.08737*, 2026.

Quan Khanh Luu, Pokuang Zhou, Zhengtong Xu, Zhiyuan Zhang, Qiang Qiu, and Yu She. Manifeel: Benchmarking and understanding visuotactile manipulation policy learning. *arXiv preprint arXiv:2505.18472*, 2025.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Yuqiang Yang, et al. Internvla-a1. 5: Unifying understanding, latent foresight, and action for compositional generalization. *arXiv preprint arXiv:2607.04988*, 2026.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Charlotte Morissette, Amin Abyaneh, Wei-Di Chang, Anas Houssaini, David Meger, Hsiu-Chin Lin, Jonathan Tremblay, and Gregory Dudek. Tactile modality fusion for vision-language-action models. *arXiv preprint arXiv:2603.14604*, 2026.

Duc Huy Nguyen, Tim Schneider, Guillaume Duret, Alap Kshirsagar, Boris Belousov, and Jan Peters. Tacex: Gelsight tactile simulation in isaac sim–combining soft-body and visuotactile simulators. *arXiv preprint arXiv:2411.04776*, 2024.

Dantong Niu, Zhuoyang Liu, Zekai Wang, Boning Shao, Zhao-Heng Yin, Anirudh Pai, Yuvan Sharma, Stefano Saravalle, Ruijie Zheng, Jing Wang, et al. T-rex: Tactile-reactive dexterous manipulation. *arXiv preprint arXiv:2606.17055*, 2026.

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

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, et al. Simfoundry: Modular and automated scene generation for policy learning and evaluation. *arXiv preprint arXiv:2606.28276*, 2026.

NeoteAI Team, Fudan TEAI Team, et al. N0-vtla: Scaling vision-tactile-language-action model with latent tactile tokens. *arXiv preprint arXiv:2607.23782*, 2026a.

NeoteAI Team, Fudan TEAI Team, et al. N0-foundation: Towards the age of tactile intelligence. *arXiv preprint arXiv:2608.29601*, 2026b.

NeoteAI Team, Fudan TEAI Team, et al. N0-twam: Scaling tactile-native world-action model for contact-rich manipulation. *arXiv preprint arXiv:2607.23783*, 2026c.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026d.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

Yi Wang, Renjun Wu, Jinyan Liu, and Xuesong Li. Vitar: Visuo-tactile residual adaptation for foundation vla manipulation. *arXiv preprint arXiv:2608.15816*, 2026.

Qiwei Wu, Rui Zhang, Xin Xiang, Tao Li, Weihua Zhang, Junjie Lai, and Renjing Xu. Tabero: Learning gentle manipulation with closed-loop force feedback from vision, touch, and language. *arXiv preprint arXiv:2605.27886*, 2026a.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026b.

Shengqi Xu, Guojin Zhong, Yang Liu, Fanjie Wang, Hu Luo, Hanyu Zhou, Weiyao Zhang, Ziyi Ye, Zuxuan Wu, and Yu-Gang Jiang. Seeing touch from motion: A unified modality-aware visuo-tactile policy with tactile motion correlation. *arXiv preprint arXiv:2606.29941*, 2026.

Zhengtong Xu, Raghava Uppuluri, Xinwei Zhang, Cael Fitch, Philip Glen Crandall, Wan Shou, Dongyi Wang, and Yu She. Unit: Data efficient tactile representation with generalization to unseen objects. *IEEE Robotics and Automation Letters*, 10(6):5481–5488, 2025.

Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022.

Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26330–26343. IEEE, 2024.

Chengbo Yuan, Zicheng Zhang, Mingjie Zhou, Wendi Chen, Yi Wang, Zhuoyang Liu, Dantong Niu, Shuo Wang, Hui Zhang, Wenkang Zhang, et al. Ftp-1: A generalist foundation tactile policy across tactile sensors for contact-rich manipulation. *arXiv preprint arXiv:2606.13102*, 2026a.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026b.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

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.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Anya Zorin, Zilin Si, Myungsun Park, Junsung Park, Alexiy Buynitsky, Sachin Bhadang, Taejun Park, Sohee John Yoon, Yong-Lae Park, Oliver Kroemer, et al. Taco: Benchmarking tactile sensors for object manipulation. *arXiv preprint arXiv:2605.21976*, 2026.

Appendix

Overview

This appendix provides additional details, experimental results, and visualizations that complement the main paper. The appendix is organized as follows:

- **Additional Related Work (Appendix A):** reviews related research on tactile learning and manipulation.
- **Limitations and Future Directions (Appendix B):** discusses the limitations of OpenViTac and potential directions for future research.
- **Benchmark and Task Details (Appendix C):** describes the tasks, sensing platforms, and success criteria.
- **Policy Implementation and Training Details (Appendix D):** documents policy implementations, tactile inputs, and training settings.
- **Additional Experimental Results (Appendix E):** reports supplementary results and focused ablations.
- **Task Visualizations (Appendix F):** illustrates tasks and tactile observations.

A Additional Related Work

A.1 Tactile Representation Learning

Early tactile representation learning often exploits paired visual–tactile observations to learn cross-modal features. Touch and Go (Yang et al., 2022) supports self-supervised learning from human-collected visual–tactile interactions, while SSVTP (Kerr et al., 2022) and Visuo-Tactile Transformers (Chen et al., 2022) further explore cross-modal representations for perception and manipulation. Subsequent work shifts toward more transferable tactile representations across sensors and tasks. UniTouch (Yang et al., 2024) aligns heterogeneous tactile observations with pretrained visual embeddings, Sparsh (Higuera et al., 2025) learns general-purpose tactile features through self-supervised pretraining, UniT (Xu et al., 2025) adopts generative reconstruction, and T3 (Zhao et al., 2025) targets transfer across heterogeneous tactile sensors and downstream tasks. More recent methods increasingly exploit the temporal structure of touch: AnyTouch2 (Feng et al., 2026) models dynamic and force-aware tactile information over short observation histories, while motion-based approaches further leverage tactile deformation dynamics to characterize evolving contact states (Xu et al., 2026). These developments motivate our comparison of generic visual, tactile-specific, and temporally-aware representations under a common policy interface.

A.2 Simulation and Real-World Robot Learning

Simulation provides a scalable way to generate interaction data and evaluate robot policies. Recent tactile platforms and benchmarks, including Tabero (Wu et al., 2026a), SoftVTBench (Jing et al., 2026), UniVTAC (Chen et al., 2026a), and NeoSim (Team et al., 2026b), further extend simulation to tactile manipulation. However, their simulated tasks are generally not constructed as direct counterparts of real-world settings, leaving differences in objects, scene layouts, task configurations, and initial conditions. A complementary line of work reduces such discrepancies through real-to-sim alignment: SIMPLER (Li et al., 2024) calibrates simulated evaluation against real-world policy behavior, RialTo (Torne et al., 2024) reconstructs digital twins for policy improvement, and SimFoundry (Ranawaka et al., 2026) automates the construction of task-relevant digital cousins. These studies highlight the importance of preserving task-relevant correspondence between simulation and the real world rather than reproducing physical scenes exactly. Recent studies further show that appropriately aligned simulated data can complement real demonstrations through sim–real co-training (Maddukuri et al., 2025; Lei et al., 2026). Motivated by these findings, we construct aligned simulated and physical tactile-manipulation tasks and study whether simulated demonstrations can improve real-world visuo-tactile policy learning.

B Limitations and Future Directions

While OpenViTac provides a unified platform for evaluating visuo-tactile manipulation across simulation and the real world, several limitations remain. The correspondence between simulated and real tactile observations is still constrained by the fidelity of tactile rendering, sensor modeling, and multimodal calibration. Future work could further improve sim-real consistency through more accurate sensor and camera calibration, while expanding benchmark coverage toward more diverse tactile interactions, including deformable object manipulation, surface interaction, and long-horizon household tasks. Moreover, extending OpenViTac to diverse robot embodiments, including dexterous hands, could support the evaluation of tactile-enabled manipulation under a broader range of contact dynamics and manipulation scenarios.

C Benchmark and Task Details

C.1 Benchmark Scope and Task Taxonomy

OpenViTac comprises 11 visuo-tactile manipulation tasks for evaluating robot policies. Following the taxonomy introduced in Sec. 3, we organize these tasks into four capability categories: physical property perception, fragility-aware manipulation, contact-rich manipulation, and precision manipulation. Tab. S1 summarizes the task abbreviations and the primary capability evaluated by each task.

Table S1: OpenViTac task taxonomy. Simulation and real-world evaluations use the same task semantics and success criteria for tasks instantiated in both domains.

Category	Abbr.	Task	Capability under evaluation
Physical Property Perception	WC	Weight Classify	Infer object weight through interaction and place the object at the corresponding target.
	HC	Hardness Classify	Infer object hardness through contact and place the object at the corresponding target.
	RC	Roughness Classify	Infer surface roughness through contact and place the object at the corresponding target.
	RR	Roughness-Guided Regrasp	Infer relative roughness and keep or regrasp the appropriate object accordingly.
	ECS	Empty-Can Selection	Distinguish visually similar empty and filled cans from their physical properties and select the instructed one.
Fragility-Aware Manipulation	GC	Grasp Chip	Grasp and place a fragile chip while avoiding damage throughout the manipulation.
Contact-Rich Manipulation	GA	Gear Assembly	Mount and rotate a gear while maintaining engagement with the paired gear.
	PD	Pull Drawer	Maintain a stable grasp on the handle while pulling the articulated drawer open.
Precision Manipulation	In-USB	Insert USB	Align and insert a USB connector into the target opening under tight geometric tolerance.
	In-B v1	Insert Block v1	Align and insert a block into the corresponding target opening.
	In-B v2	Insert Block v2	Select the instructed object and then align and insert it into the corresponding target opening.

C.2 Simulation and Real-world Environment

OpenViTac builds scalable visuo-tactile simulation environments with corresponding real-world task instantiations for consistent policy evaluation. The simulation environments are developed upon NVIDIA Isaac Sim 5.1.0 and Isaac Lab 2.3.0, with tactile observations generated using TacEx-based tactile simulation (Nguyen et al., 2024). The platform supports three representative tactile sensors,

including GelSight, Xense, and NeoteAI, together with two widely used gripper configurations: the Franka Hand and Robotiq gripper.

To generate realistic visuo-tactile observations, OpenViTac separates visual asset construction from tactile simulation. While TacEx provides physics-based tactile rendering through contact interaction, it does not directly model the full visual appearance of objects, such as diverse textures and colors. Therefore, we construct visually realistic object assets independently and associate them with TacEx-compatible tactile representations using shared object geometry. This design enables consistent RGB and tactile observations from the same simulated objects.

For real-world evaluation, we instantiate the corresponding tasks using the same robot embodiments, gripper configurations, tactile sensors, and task objectives as in simulation. Such correspondence preserves consistent interaction semantics between simulation and reality, enabling meaningful comparison of policies across both environments.

To facilitate policy deployment and comparison across different robot learning frameworks, OpenViTac provides standardized robot state and action interfaces. Both end-effector-space and joint-space representations are supported, including multiple rotation parameterizations such as quaternions, rotation matrices, and rotation vectors. These representations are consistently provided in both simulation and real-world environments, enabling compatibility with pretrained manipulation policies that adopt different action and state formulations.

C.3 Data Collection Statistics

OpenViTac contains both simulated trajectories and real-world robot demonstrations collected under three tactile sensing configurations: GelSight, Xense, and Neote. The collected data statistics are summarized in Table S2.

For each tactile sensor configuration, the simulation benchmark contains 1,300 episodes across 11 tasks, corresponding to approximately 918K recorded frames in total. The real-world dataset contains 2,400 demonstrations per tactile sensor configuration across 8 tasks, with approximately 2.1M recorded frames collected in total.

To improve task diversity, several tasks include multiple object configurations or language-conditioned variations. For example, Insert Block v2 contains three object semantics with two scene layouts, resulting in six data collection configurations in the real-world setting. Similarly, simulation tasks with binary physical properties, such as weight, hardness, and roughness classification, are balanced across different object conditions.

Table S2: Data collection statistics of OpenViTac. Statistics are reported per tactile sensor configuration for both simulated episodes and real-world demonstrations.

(a) Simulation

Task	Episodes
Weight Classify (WC)	100
Hardness Classify (HC)	100
Roughness Classify (RC)	100
Roughness-Guided Regrasp (RR)	100
Empty-Can Selection (ECS)	100
Grasp Chip (GC)	100
Gear Assembly (GA)	100
Pull Drawer (PD)	100
Insert USB (In-USB)	100
Insert Block v1 (In-B v1)	100
Insert Block v2 (In-B v2)	300
Total	**1,300**

(b) Real-world

Task	Demonstrations
Roughness Classify (RC)	100
Empty-Can Selection (ECS)	100
Grasp Chip (GC)	100
Gear Assembly (GA)	200
Pull Drawer (PD)	100
Insert USB (In-USB)	200
Insert Block v1 (In-B v1)	400
Insert Block v2 (In-B v2)	1200
Total	**2,400**

C.4 FULL TASK CATALOG

Table S3 provides detailed descriptions of the 11 manipulation tasks in OpenViTac. Each task is designed to evaluate a specific tactile-relevant capability while maintaining consistent task objectives across simulation and real-world settings. The success conditions are defined based on task completion requirements, including object states, interaction outcomes, and physical constraints when applicable.

Table S3: Task catalog of OpenViTac, including task objectives and high-level success conditions.

Task	**Task Objective**	**Success Condition**
Weight Classify (WC)	Infer whether a block is light or heavy through interaction and place it on the corresponding target.	The block is correctly classified and placed on the corresponding target plate.
Hardness Classify (HC)	Infer whether a block is soft or hard through contact interaction and place it on the corresponding target.	The block is correctly classified and placed on the corresponding target plate.
Roughness Classify (RC)	Infer whether a block is smooth or rough through contact interaction and place it on the corresponding target.	The block is correctly classified and placed on the corresponding target plate.
Empty-Can Selection (ECS)	Identify the empty can among visually similar cans through physical interaction and place it into the basket.	The empty can is correctly identified and successfully placed into the basket.
Roughness-Guided Regrasp (RR)	Determine whether to retain the initially grasped block based on roughness and regrasp when necessary.	The rough block is correctly selected and placed on the target plate.
Grasp Chip (GC)	Grasp and transport a fragile chip while avoiding damage during manipulation.	The chip is successfully grasped, transported, and placed without damage while satisfying physical safety constraints.
Gear Assembly (GA)	Assemble a gear onto the base and rotate it to drive the paired gear through physical engagement.	The gears are successfully assembled and achieve coupled rotation through stable mechanical interaction.
Pull Drawer (PD)	Grasp the drawer handle and pull the drawer open while maintaining stable contact.	The drawer is successfully grasped and pulled open with the required displacement.
Insert USB (In-USB)	Align the USB connector with the target slot and complete insertion.	The USB connector is successfully aligned and inserted into the target slot.
Insert Block v1 (In-B v1)	Grasp the designated cube and insert it into the matching opening under different scene layouts.	The designated block is successfully aligned and inserted into the corresponding opening.
Insert Block v2 (In-B v2)	Select the language-specified object and insert it into the corresponding opening across different layouts.	The instructed object is correctly selected and successfully inserted into the matching opening.

D Policy Implementation and Training Details

We train a separate policy for each task. For simulation, we train on 100 demonstration episodes per task, except Insert Block v2, which uses 300 episodes. For real-world training, we randomly select 100 episodes per task from the larger collected demonstration pool, or 300 episodes for Insert Block v2. The evaluated VLA, WAM, and VTLA policies are listed in Tab. S4. For previously published models, we use each method's prescribed initialization and retain its official training configuration except for the training steps and global batch size specified in the table. FastWAM, Lingbot-VA, and StarVLA (Cosmos) follow the LIBERO configuration in their respective official repositories; N0-TWAM follows its official repository's training configuration. OpenVTLA uses the same implementation as $\pi_{0.5}$. Model-specific preprocessing and action representations follow the corresponding source recipe, with adapters used to train and execute each policy on the benchmark tasks.

Table S4: Training settings for the evaluated policies.

	Models	Steps	Global batch
VLAs	$\pi_{0.5}$	20k	64
	StarVLA (Qwen3)	20k	64
	InternVLA-A1.5	40k	32
	XVLA	30k	64
	Lingbot-VLA-2	20k	64
	Xiaomi-Robotics	20k	64
WAMs	FastWAM	20k	8
	Lingbot-VA	5k	8
	StarVLA (Cosmos)	20k	8
	N0-TWAM	2k	16
VTLAs	FTP-1	20k	64
	N0-VTLA	20k	64
	OpenVTLA (ours)	20k	64

Training steps count optimizer updates, and global batch size includes any gradient accumulation. The native optimizer, learning-rate schedule, and prediction horizon follow the source recipe unless adaptation to the policy's action representation is required. Because the architectures and training recipes differ across families, cross-family results compare complete policies rather than isolating the effect of a single training choice. We evaluate the final checkpoint of each training run instead of selecting a checkpoint by validation performance.

E Additional Experimental Results

We further examine two factors that may contribute to the effectiveness of tactile representations: tactile-specific pretraining and temporal context. All variants retain the same OpenVTLA architecture and token-level integration interface, changing only the factor under study.

Effect of tactile pretraining. Tab. S5 compares the pretrained AnyTouch2 encoder with an identically structured encoder initialized from scratch. Removing tactile pretraining reduces the overall performance from 68.7% to 56.5%, with consistent degradation across all four capability groups. The largest drop occurs on Fragility-Aware (FA) tasks, while substantial decreases are also observed on Property Perception (PP) and Contact-Rich (CR) tasks. These results indicate that tactile-specific pretraining provides representations that transfer effectively to downstream VLA adaptation.

Table S5: Comparison between pretrained AnyTouch2 and its randomly initialized counterpart across the four capability groups.

Representation	**PP**	**FA**	**CR**	**PR**	**Avg.**
AnyTouch2 (Random Init.)	78.5	34.0	82.0	10.2	56.5
AnyTouch2 (Pretrained)	**90.2**	**69.0**	**95.0**	**15.3**	**68.7**

Effect of temporal context. We next remove temporal variation while preserving the encoder and policy architecture by replacing the tactile history with repeated copies of the current frame. As shown in Tab. S6, this decreases the overall performance from 68.7% to 58.4%. The degradation is most pronounced on PP and FA tasks, where observations of evolving contact can provide information about physical properties and interaction states. Similar trends are also observed when introducing temporal tactile context to SigLIP and T3, suggesting that the benefit of temporal information is not specific to AnyTouch2. Overall, these results indicate that tactile dynamics provide useful information beyond static contact appearance, particularly for tasks that depend on physical-property and contact-state estimation.

Table S6: Comparison between the original tactile history and repeated copies of the current frame across the four capability groups.

Temporal Input	**PP**	**FA**	**CR**	**PR**	**Avg.**
Repeated Current Frame	79.0	52.0	90.0	5.2	58.4
Tactile History	**90.2**	**69.0**	**95.0**	**15.3**	**68.7**

F TASK VISUALIZATIONS

F.1 VISUALIZATIONS ACROSS TACTILE SENSORS

Fig. S1 compares representative observations from the insert USB task with GelSight, Xense, and NeoteAI sensors. Each panel shows camera views together with left- and right-fingertip tactile readings at selected stages of the interaction. The comparison illustrates how the same task objective appears through different tactile sensing modalities.

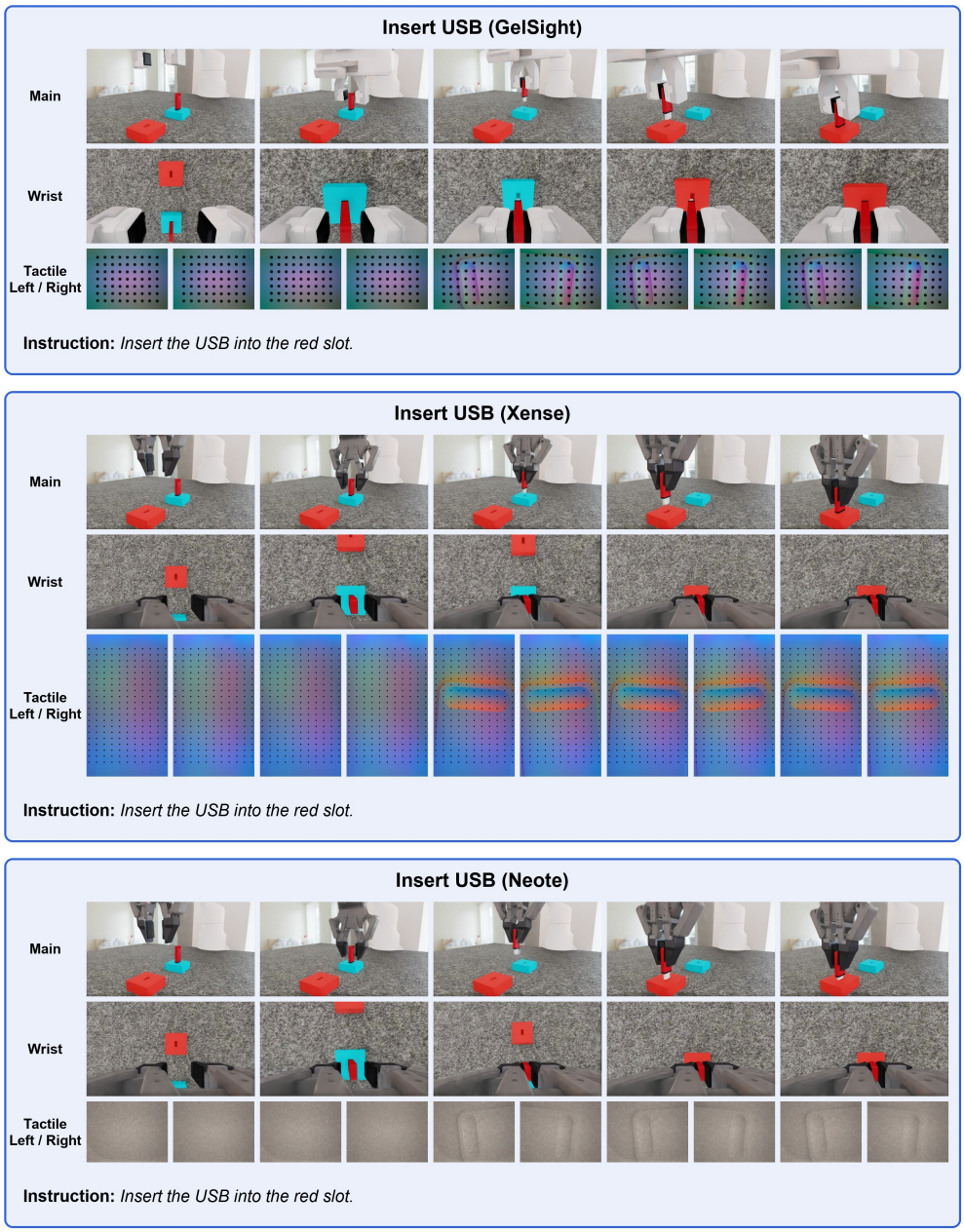


Figure S1: Visualization of the *insert USB* task using GelSight, Xense, and NeoteAI tactile sensors.

F.2 SIMULATED TASKS

Fig. S2–S13 visualize all 11 simulated tasks across physical-property perception, fragility-aware manipulation, contact-rich manipulation, and precision manipulation. The panels place main-camera, wrist-camera, and bilateral tactile observations in temporal order; multi-case figures additionally show distinct task conditions or object variants. These examples make the interaction sequence and the contact information available to the policy easier to inspect.

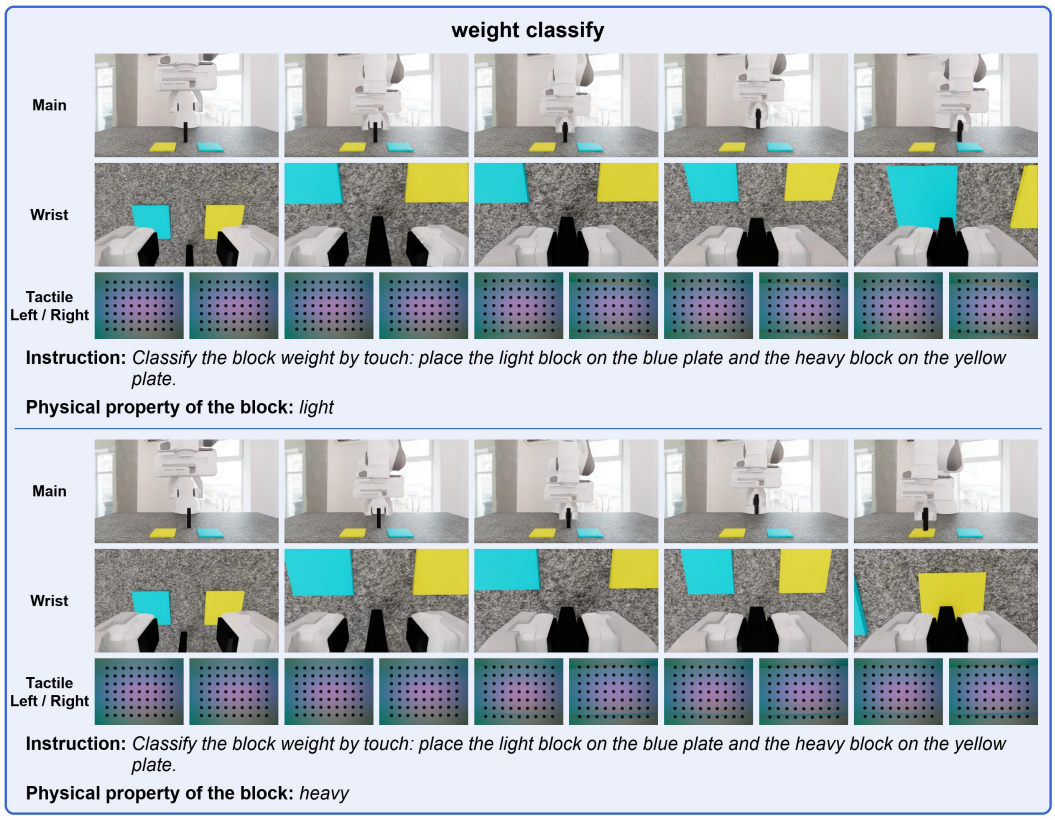


Figure S2: Visualization of *weight classify* task.

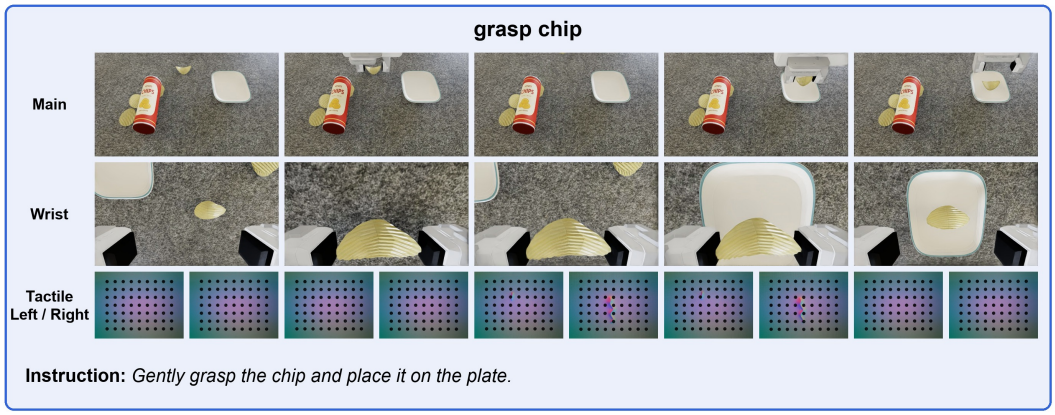


Figure S3: Visualization of *grasp chip* task.

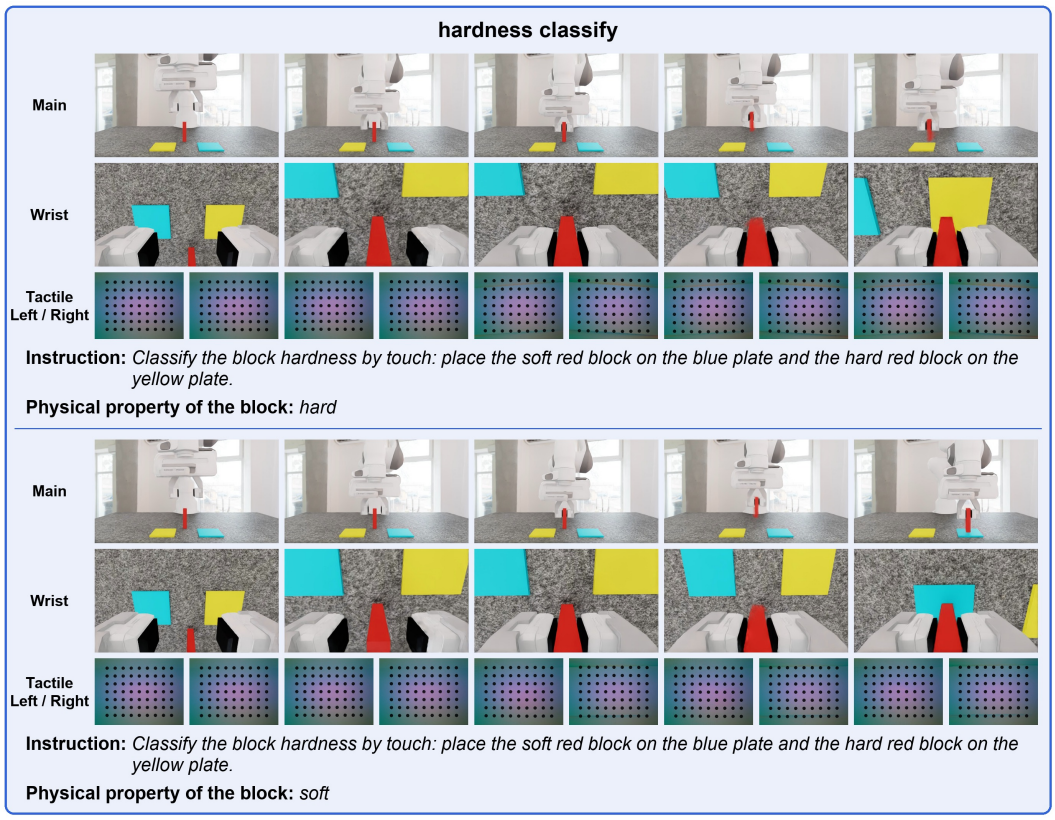


Figure S4: Visualization of *hardness classify* task.

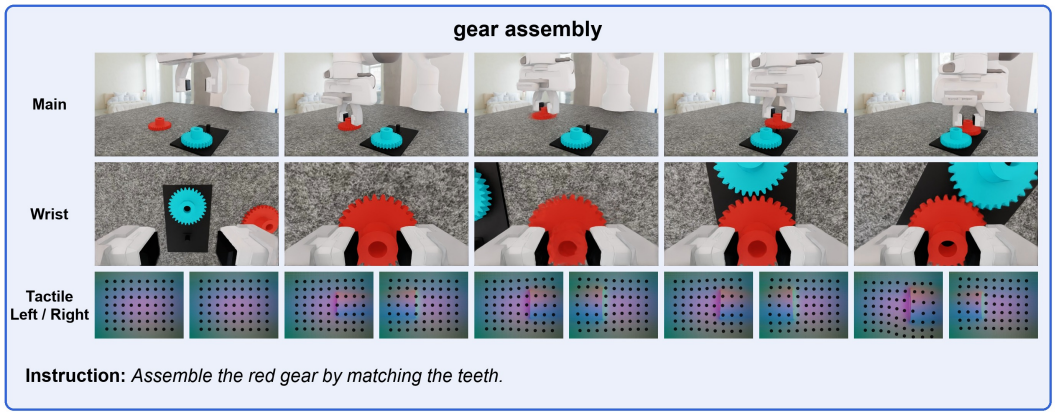


Figure S5: Visualization of *gear assembly* task.

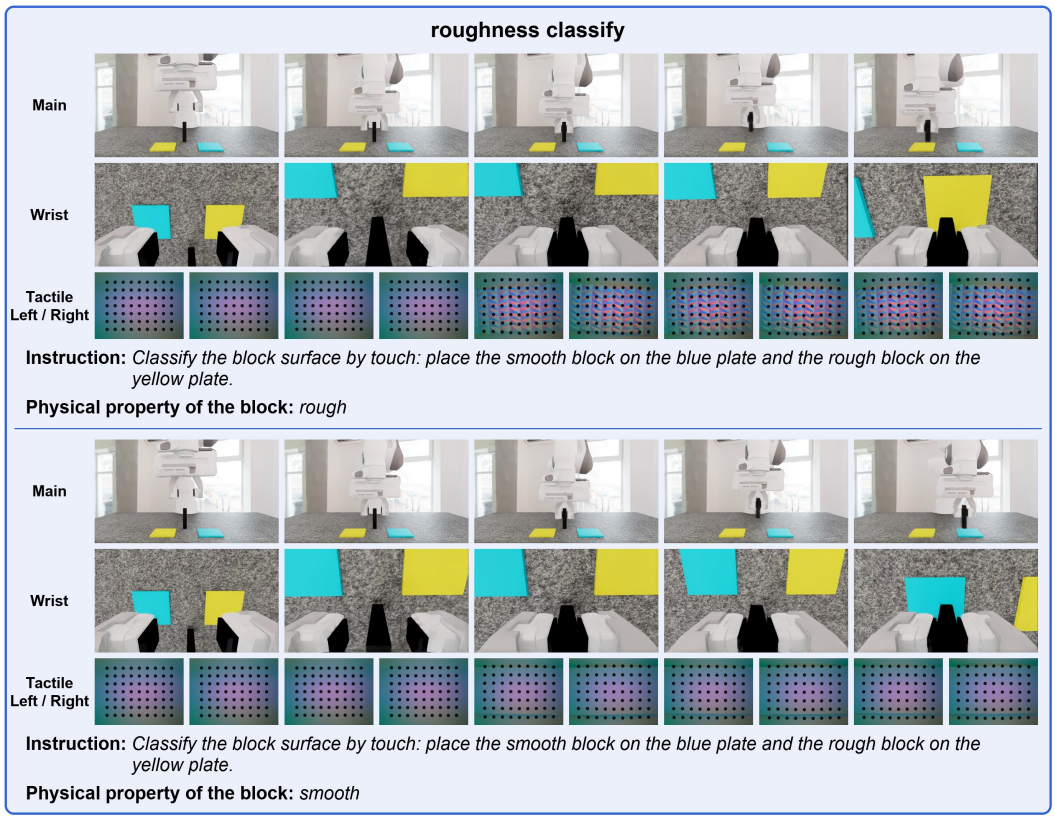


Figure S6: Visualization of *roughness classify* task.

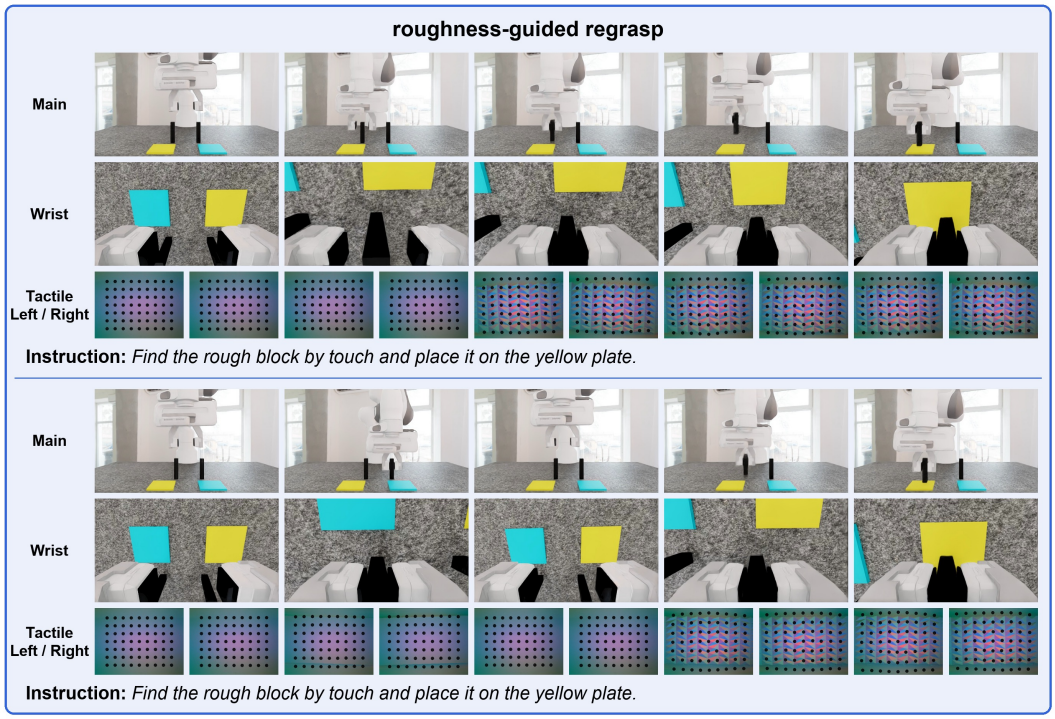


Figure S7: Visualization of *roughness-guided regrasp* task.

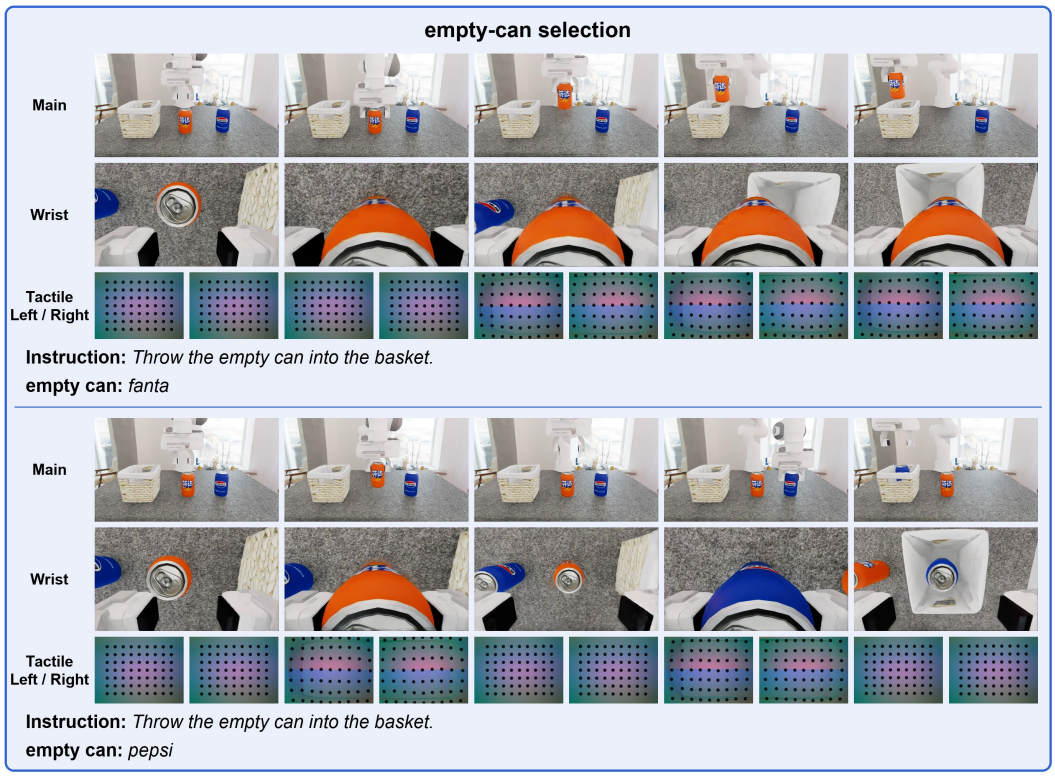


Figure S8: Visualization of *empty-can selection* task.

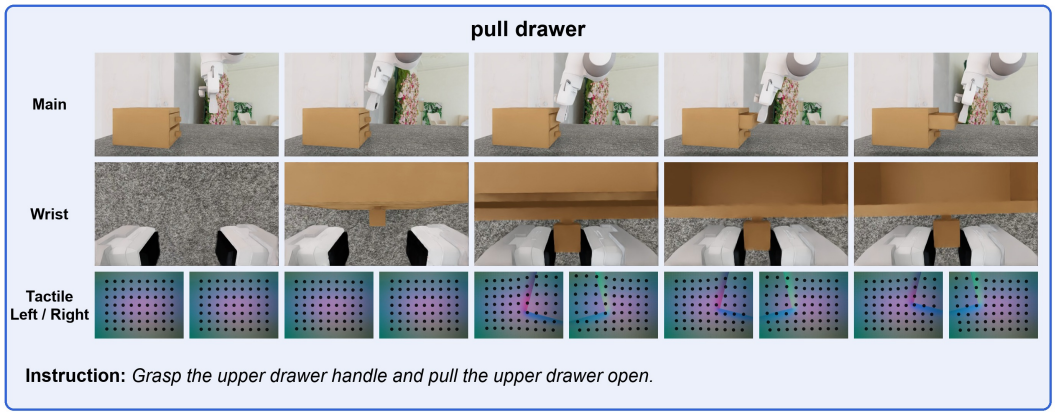


Figure S9: Visualization of *pull drawer* task.

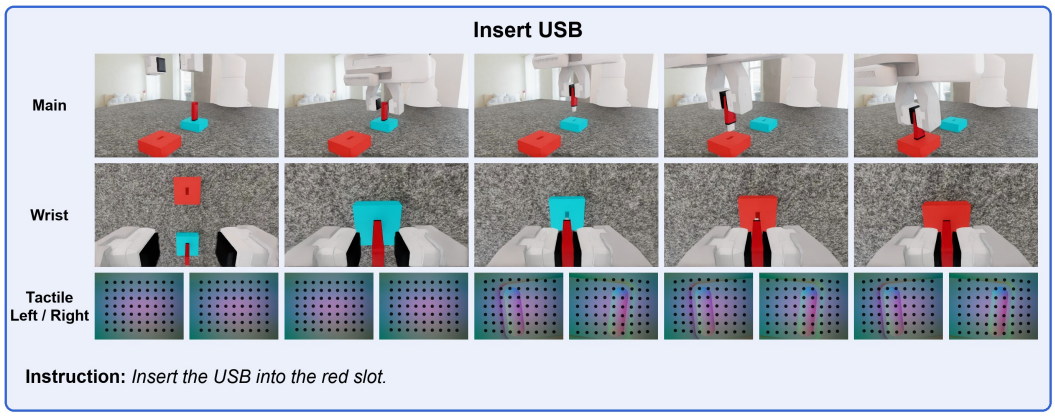


Figure S10: Visualization of *insert USB* task.

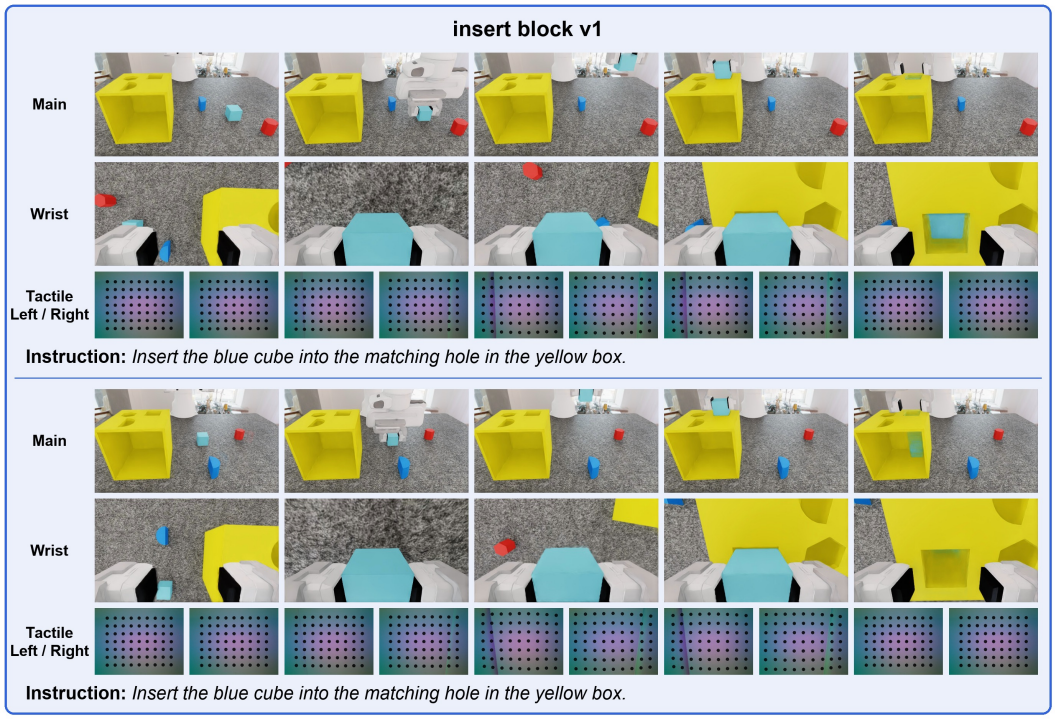


Figure S11: Visualization of *insert block v1* task.

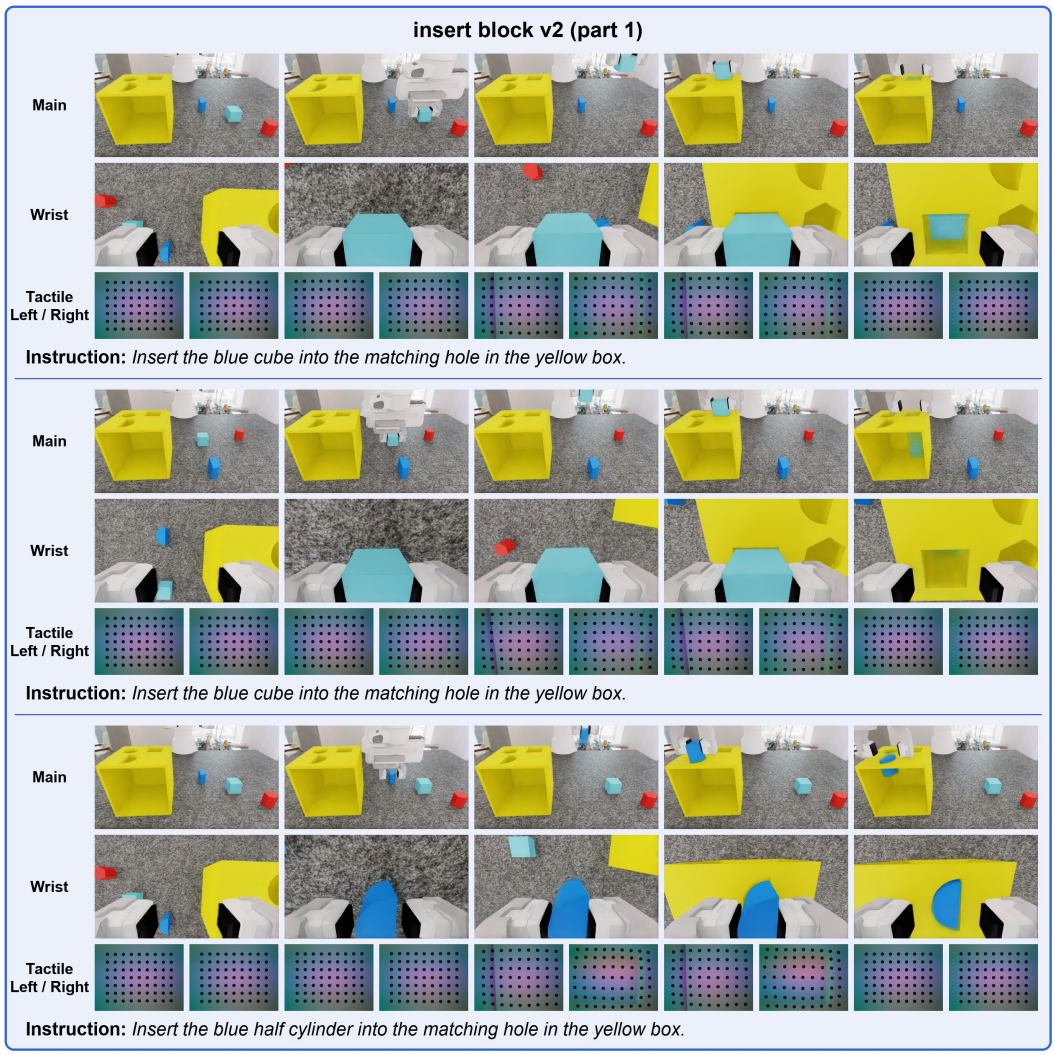


Figure S12: Visualization of *insert block v2* task (part 1).

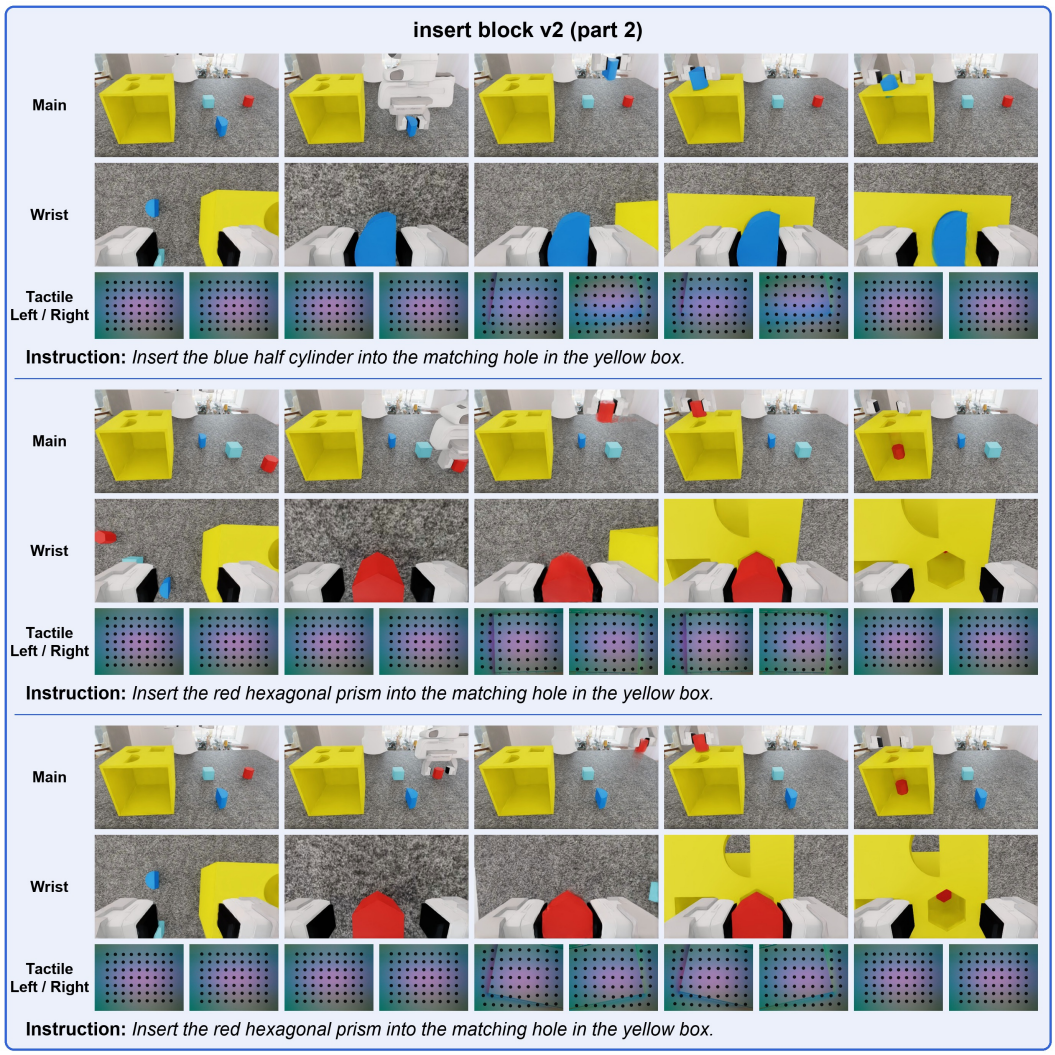


Figure S13: Visualization of *insert block v2* task (part 2).

F.3 Real-World Tasks

Fig. S14–S21 show representative real-world instances of the paired benchmark tasks. As in the simulated figures, the camera and fingertip tactile views are arranged over successive interaction stages. The examples provide qualitative context for object selection, gentle grasping, sustained contact, and precise insertion with physical objects and sensors.

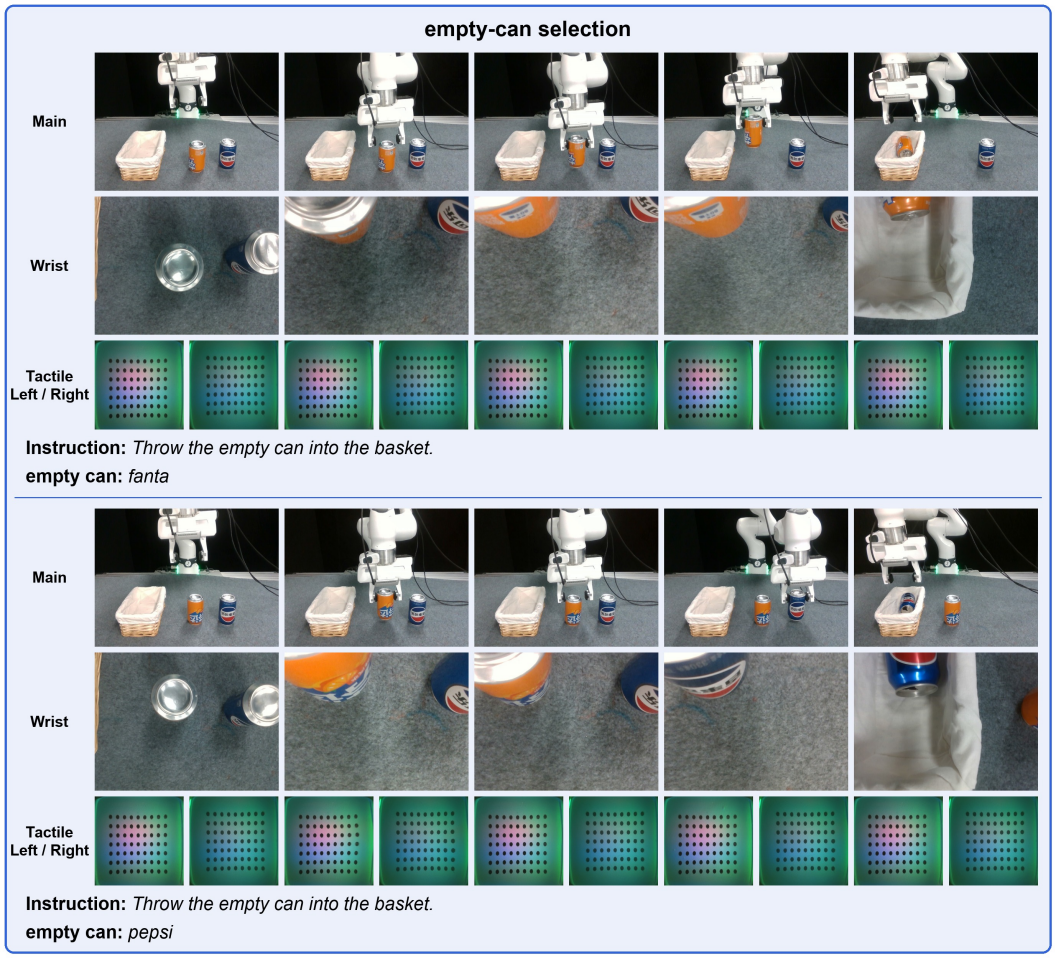


Figure S14: Visualization of *empty-can selection* task.

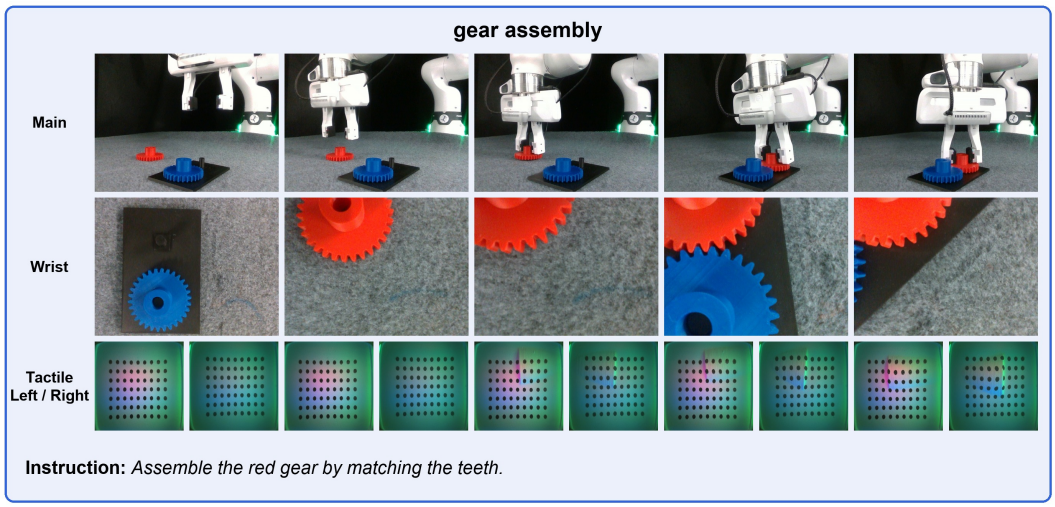


Figure S15: Visualization of *gear assembly* task.

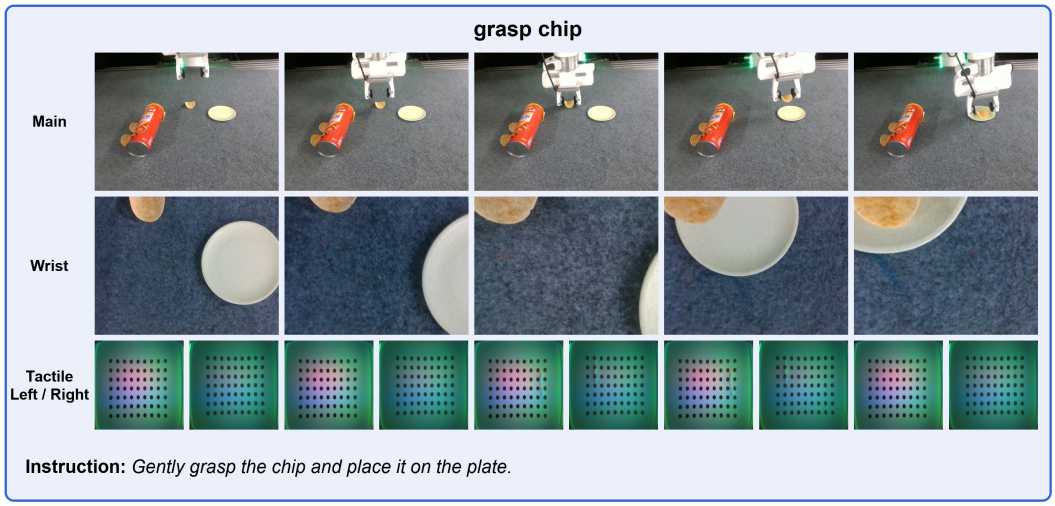


Figure S16: Visualization of *grasp chip* task.

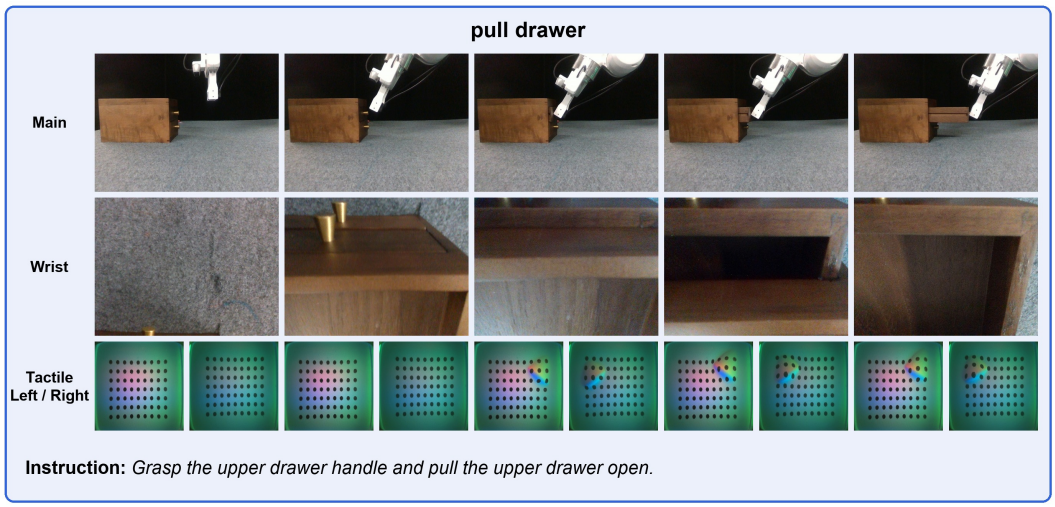


Figure S17: Visualization of *pull drawer* task.

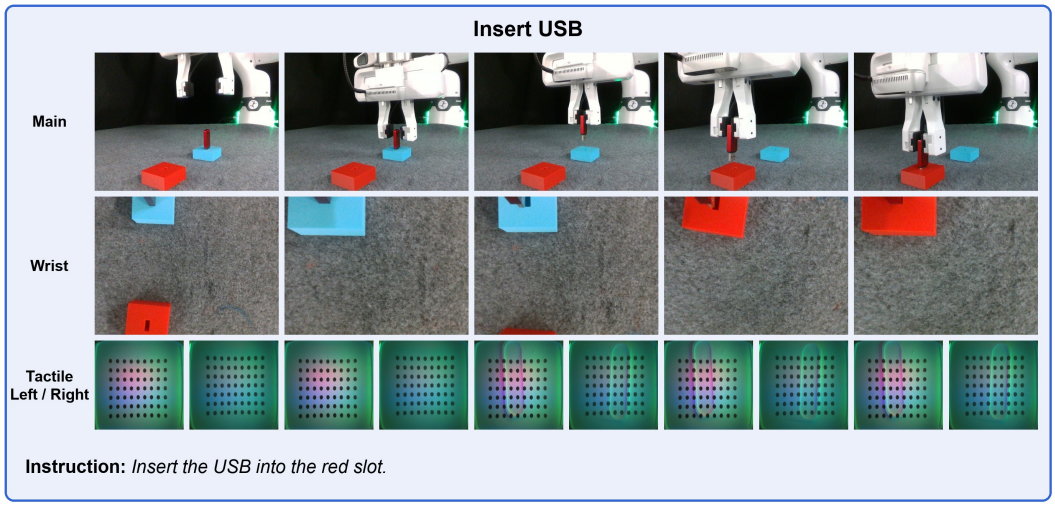


Figure S18: Visualization of *insert USB* task.

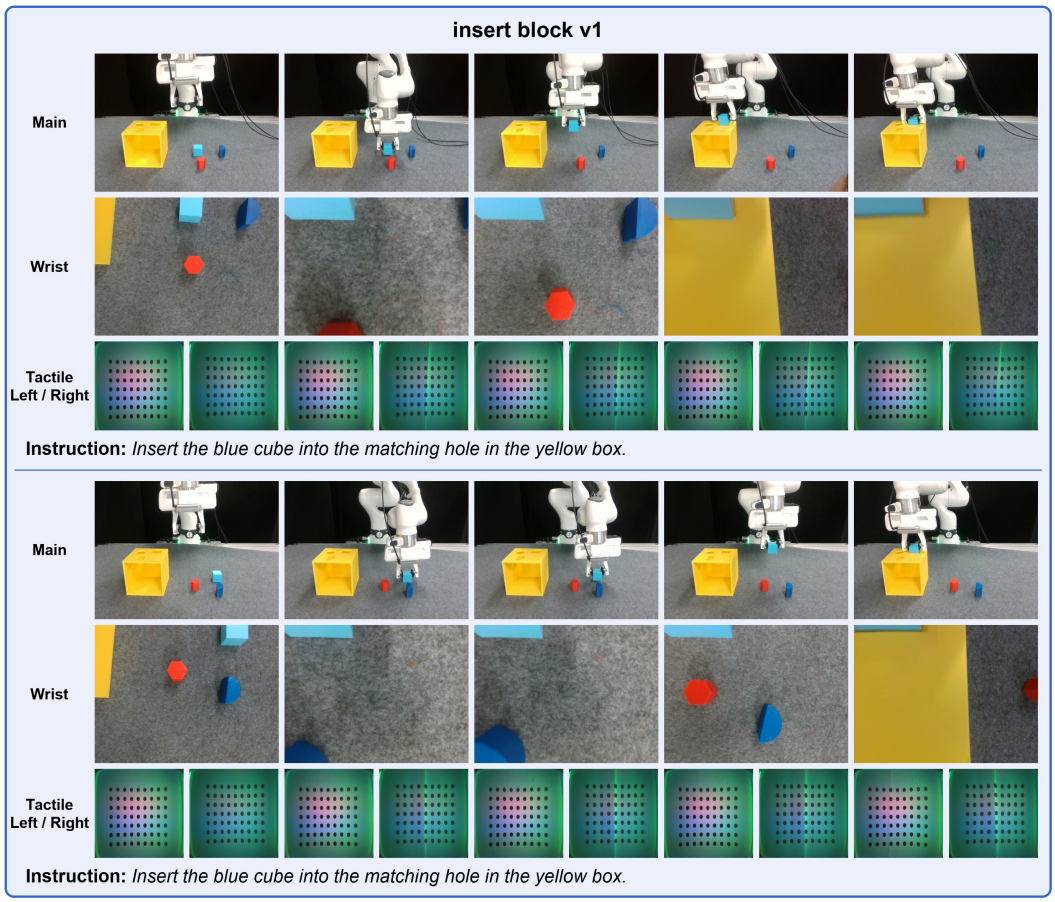


Figure S19: Visualization of *insert block v1* task.

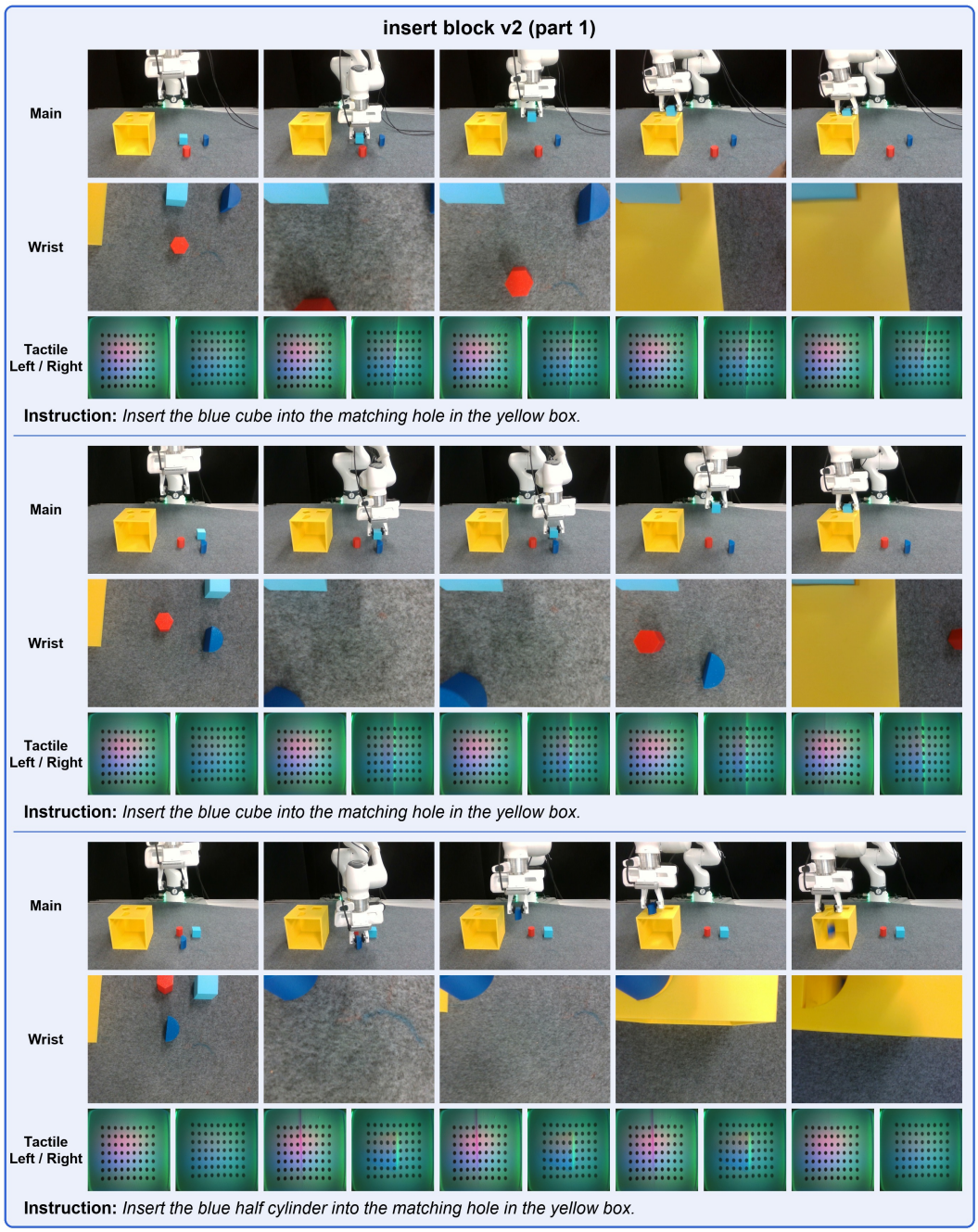


Figure S20: Visualization of *insert block v2* task (part 1).

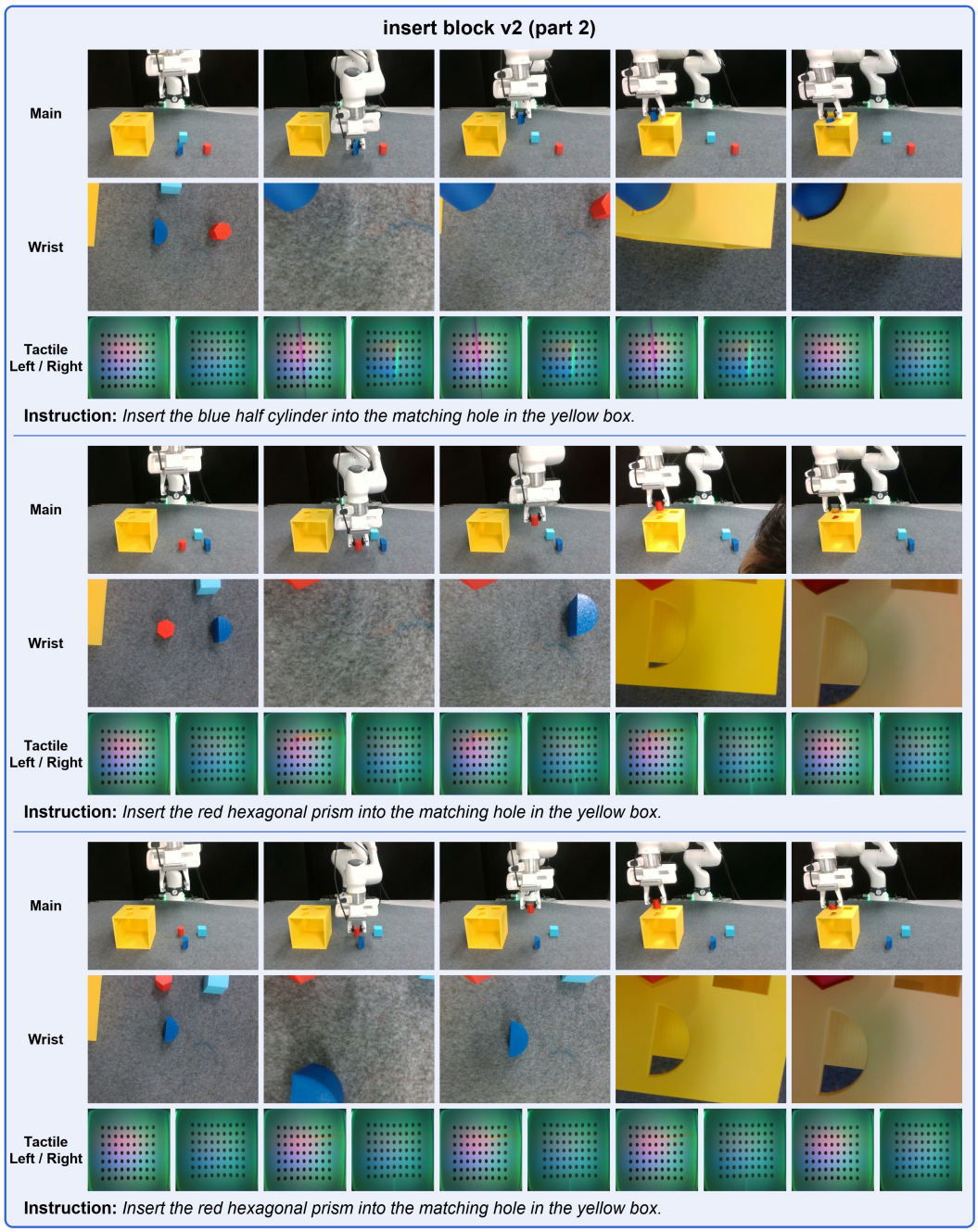


Figure S21: Visualization of *insert block v2* task (part 2).