

TacCoRL: Integrating Tactile Feedback into VLA via Simulation

Anonymous
Affiliation

Abstract: Vision-language-action (VLA) models provide strong visual, language, and action priors for robot manipulation, but visual observations alone often miss the local contact state required for contact-rich tasks. We present **TacCoRL**, a scalable framework that injects **Tactile** feedback into VLA policies and improves them through sim-real **Co-training** and simulation-based reinforcement learning (**RL**), without requiring large-scale tactile pretraining or extensive real-world contact exploration. The key idea is not only adding touch as an input, but learning how contact readings should modulate action responses in near-failure states that are rare in demonstrations and risky to collect on hardware. We use a real-aligned simulator as a closed-loop training environment for contact interaction. Mixed simulated and real trajectories first warm-start tactile-conditioned actions in the pretrained policy. Reinforcement learning with verifiable task rewards then optimizes the policy using simulated contact rollouts. It reinforces tactile-conditioned actions that lead to task completion, while a supervised objective on real trajectories keeps the refined policy anchored to deployment visual, tactile, and action distributions. The resulting policy transfers directly to the real robot without privileged simulation state or online real-world RL. Across four bimanual contact-rich tasks, the final visuo-tactile policy achieves an average success rate of 72.5%, compared to baseline of 50.0%. Result videos and more details are available at <https://tac-corl.github.io/>.

Keywords: Tactile Feedback, Vision-Language-Action Models, Sim-to-Real Post-Training, Reinforcement Learning, Contact-Rich Manipulation

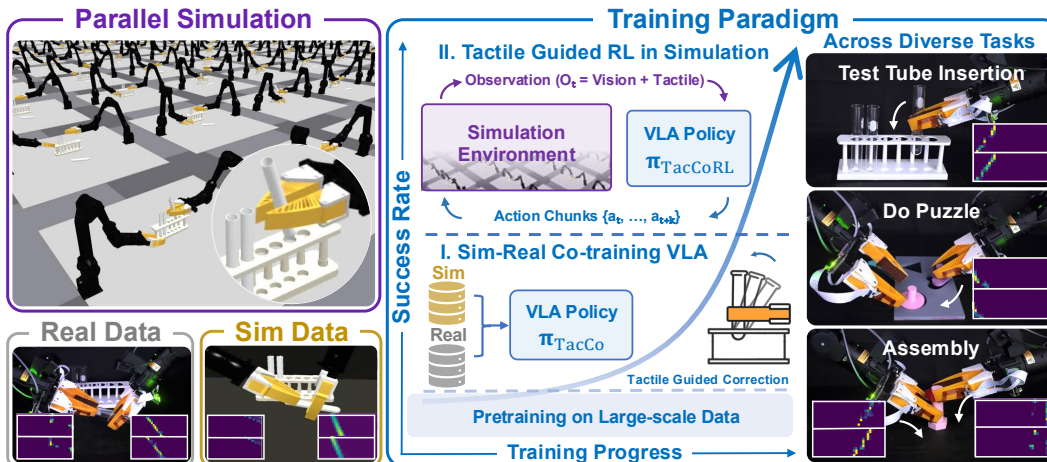


Figure 1: *Left:* We collect real and simulated visuo-tactile trajectories from aligned real-world and simulation setups. *Center:* Sim-real co-training gives the policy an initial tactile-conditioned action prior, and tactile-guided RL in a real-aligned simulator refines closed-loop contact corrections. *Right:* We deploy the policy directly to the real world, where it achieves high success rates across diverse contact-rich manipulation tasks.

1 Introduction

Precise contact-rich manipulation requires vision to interpret the scene and touch to resolve interactions at the contact interface. Vision-Language-Action (VLA) models provide strong visual

representations, language grounding, and action priors from large-scale pretraining data [1, 2, 3, 4], making them a compelling foundation for robotic manipulation. However, in contact-rich tasks such as insertion, assembly, and in-hand manipulation, cameras often cannot reliably observe whether the manipulated part is properly aligned, where contact occurs, or how pressure is distributed. Tactile sensing [5, 6, 7, 8] provides this complementary signal by revealing local contact location, pressure distribution, and misalignment. While recent work develops tactile capabilities through large tactile datasets [9, 10, 11], tactile-language-action models [12, 13, 14, 15, 16, 17], or inference-time tactile guidance [18], we investigate whether tactile feedback can instead be *integrated into an existing VLA backbone* without large-scale tactile-labeled pretraining.

Real demonstrations alone are ill-suited for this tactile-aware injection problem. They are costly to scale, emphasize successful nominal behavior, and underrepresent the near-failure cases where tactile feedback is most critical: slight misalignment, contact on the wrong surface, and unstable grasp-object interactions. Imitation learning can expose the policy to tactile observations, but it provides limited supervision on how contact should influence actions in off-nominal states; collecting such cases directly on hardware risks sensor damage and incurs slow and costly resets. We therefore require a safe, resettable, and verifiable simulated tactile environment [19, 20, 8, 21], where contact conditions can be systematically varied and learned from in closed loop.

We use simulation as the tactile-learning environment, aligned with the real task setup through matched scene configuration, robot controller response, and tactile-contact interfaces calibration [8, 22]. Mixed sim-real co-training [23, 24] provides the pretrained VLA [3, 4] with an initial tactile-conditioned action prior. Static demonstrations cannot teach the policy to recover from its own contact-induced errors. We therefore apply sparse-reward reinforcement learning (RL) in simulation to reinforce successful contact behaviors [25, 26, 27, 28, 29], while a supervised real-data anchor keeps the refined policy aligned with real-world distribution statistics.

TacCoRL treats tactile integration as an efficient sim-to-real post-training framework for pretrained VLA policies (see Figure 1). Our framework offers three key contributions: (1) a tactile-conditioning interface that adapts pretrained VLA policies without requiring large-scale tactile pretraining or training tactile policies from scratch; (2) a sim-to-real post-training pipeline that combines sim-real co-training with verifiable rewards in simulation, anchored by real data to learn transferable contact-guided action refinement; and (3) real-robot evaluations on four bimanual contact-rich manipulation tasks, demonstrating improved success and robustness over vision- and imitation-only baselines.

2 Related Work

Tactile-Augmented VLA Policies. Vision-language-action models provide strong pretrained priors, but widely used backbones are still trained mainly on vision, language, and proprioception [1, 2, 3, 4]. Contact-rich manipulation often depends on fine-grained contact geometry, local contact states, and subtle force variations that are occluded or visually ambiguous, motivating tactile sensing as a complementary modality [5, 30, 6, 31, 32, 33]. Prior work develops tactile capability through tactile-language-action models [12, 13, 14], large tactile datasets [9, 10], cross-sensor tactile representations [11, 34, 35], and sensor-vision-language alignment objectives [36, 37], or by augmenting existing VLA policies with tactile inputs [7, 16], tactile-aware modules [15, 17, 38], or inference-time guidance [18]. Our work follows the augmentation view: it injects tactile feedback into an existing VLA backbone rather than relying on large-scale tactile pretraining or training a tactile policy from scratch.

Sim-Real Post-Training for Contact-Rich Policies. Post-training refines learned policies beyond demonstrations, which is useful when important contact states are rare in supervised data [29]. Recent VLA work studies online post-training with simulated or real-world rollouts [25, 26, 39], buffered experience and scalable RL training [27, 40, 41], offline advantage-based updates [42], and sim-real regularization [28]. Related work outside VLAs applies RL to diffusion policies [43, 44, 45, 46], residual controllers [47, 48], shared-autonomy controllers [22], and simulation-

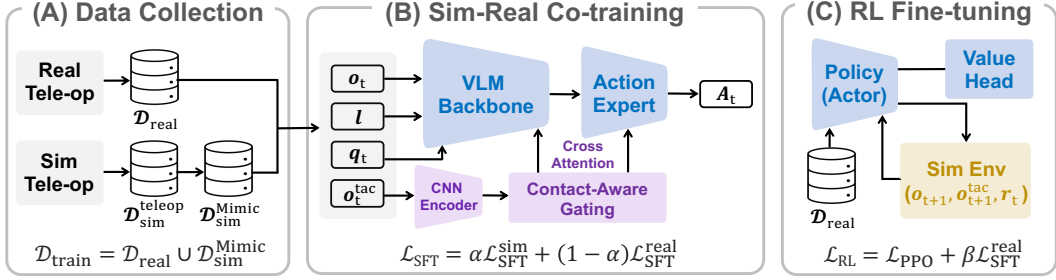


Figure 2: **Pipeline.** (A) We collect real demonstrations $\mathcal{D}_{\text{real}}$ together with simulated tele-operation data $\mathcal{D}_{\text{sim}}^{\text{teleop}}$, and further scale up $\mathcal{D}_{\text{sim}}^{\text{teleop}}$ using MimicGen to obtain $\mathcal{D}_{\text{sim}}^{\text{Mimic}}$. (B) During sim-real co-training, tactile information is encoded and routed through contact-aware gating to modulate both the context of vision-language models (VLM) and the action expert. (C) Interactive simulation rollouts help the tactile-aware policy improve closed-loop task performance, while a supervised real-data loss constrains the update to avoid drifting away from the real robot distribution.

based dexterous manipulation [8, 49, 50]. Our method develops a tactile-aware sim-to-real post-training procedure for adapting existing VLA policies to contact-rich manipulation.

3 Method

Our framework adapts a pretrained VLA model to incorporate tactile feedback via a two-stage sim-to-real pipeline. Mixed sim-real co-training first initializes the pretrained VLA with a tactile-conditioned action prior learned from both simulation and real-world expert rollouts. Sparse-reward RL further refines the policy through interactive simulation rollouts, while real-data supervision regularizes the update toward the real-world distribution. At time t , the policy receives a language instruction ℓ , images \mathbf{o}_t^v , proprioception \mathbf{q}_t , and tactile readings \mathbf{o}_t^τ , and predicts an H -step action chunk $\mathbf{A}_t = \mathbf{a}_{t:t+H-1} \in \mathbb{R}^{H \times d_a}$, where d_a denotes the action dimension. We denote the actor observation by

$$\mathbf{x}_t = (\ell, \mathbf{o}_t^v, \mathbf{q}_t, \mathbf{o}_t^\tau), \quad \mathbf{A}_t \sim \pi_\theta(\cdot | \mathbf{x}_t). \quad (1)$$

The real robot environment is E_{real} , and the simulation environment is $E_{\text{sim}}(\psi)$ with alignment parameters ψ . Both domains expose \mathbf{x}_t ; the simulator additionally exposes privileged state \mathbf{s}_t for reward and critic learning. Figure 2 summarizes the overall pipeline.

3.1 Simulation Environment Alignment

Our method uses simulation as a closed-loop contact environment, rather than solely as an offline data source. To make simulation rollouts effective for policy learning, we align the policy-facing interfaces between simulation and the real-world setup: scene configuration and camera define the visual task context; controller response determines how VLA action chunks change the robot state and affects transfer [51, 52]; and tactile statistics determine how contact occurrence, location, magnitude, and temporal variation are presented to the policy [8, 20, 21]. Details of controller identification, tactile-signal alignment, and camera alignment are provided in Appendix B.

3.2 Tactile-Augmented VLA Policy

We instantiate the policy using a $\pi_{0.5}$ -style VLA model [4]: a strong vision-language backbone provides semantic and visual context, while an action expert generates action chunks via a flow policy [3, 4]. The base VLA tokenizes language, images, and proprioception into

$$\mathbf{Z}_t^{\text{base}} = [\mathbf{Z}^\ell, \mathbf{Z}_t^v, \mathbf{Z}_t^q]. \quad (2)$$

Our tactile interface adds a tactile encoder E_τ and a projection \mathbf{W}_τ that produce M tactile tokens of hidden size d . Motivated by OpenTouch’s contact-history analysis [53], E_τ encodes a recent tactile window $\mathbf{h}_t^\tau = \mathbf{o}_{t-L+1:t}^\tau \in \mathbb{R}^{L \times K}$ with history length L and K taxels, capturing how contact is

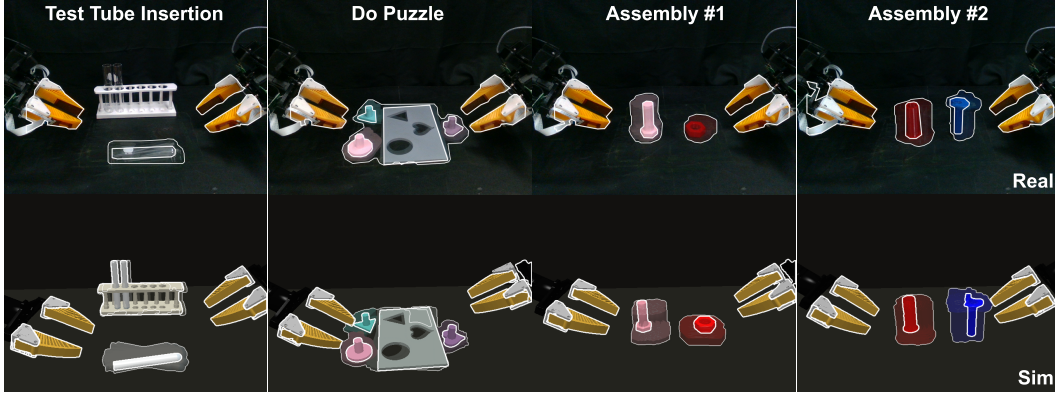


Figure 3: **Experimental task settings.** Real and calibrated simulation workspaces for four contact-rich bimanual tasks. The accumulated object placements indicate the pose ranges used for domain randomization and evaluation.

loaded, released, and evolves over time:

$$\mathbf{Z}_t^\tau = \mathbf{W}_\tau E_\tau(\mathbf{h}_t^\tau) \in \mathbb{R}^{M \times d}. \quad (3)$$

A binary contact gate suppresses the influence of tactile tokens when it contains only background noise. For contact-aware tactile fusion [17], we compute the gate over taxel readings $f_{r,k}$ with activation threshold λ_f and active-count threshold m :

$$c_t = I \left[\max_{r \in [t-L+1, t]} \sum_{k=1}^K I(|f_{r,k}| > \lambda_f) \geq m \right], \quad \tilde{\mathbf{Z}}_t^\tau = \text{Gate}(\mathbf{Z}_t^\tau, c_t). \quad (4)$$

The gated tactile tokens enter the model through two conditioning paths:

$$\bar{\mathbf{Z}}_t^{\text{base}} = \text{CrossAttn}(\mathbf{Z}_t^{\text{base}}, \tilde{\mathbf{Z}}_t^\tau), \quad \mathbf{Z}_t = [\bar{\mathbf{Z}}_t^{\text{base}}, \tilde{\mathbf{Z}}_t^\tau]. \quad (5)$$

Here, $\text{Gate}(\mathbf{Z}_t^\tau, c_t)$ returns \mathbf{Z}_t^τ under contact and removes tactile tokens from attention otherwise. The first path allows tactile history to update the vision-language-proprioception context; the second keeps tactile tokens available as conditioning tokens for the action expert at each denoising step. This design maintains the pretrained VLA behavior outside contact while allowing tactile evidence to reshape the action chunk during contact.

3.3 Sim-Real Co-Training

The tactile-augmented VLA provides an interface for touch, but the pretrained backbone has not learned how tactile events should influence actions. We first warm-start the policy via mixed supervised co-training. Let

$$\mathcal{D}_\rho = \{(\mathbf{x}_t^i, \mathbf{A}_t^{i,*})\}_{i,t}, \quad \rho \in \{\text{real}, \text{sim}\}, \quad (6)$$

where both datasets share the same representation \mathbf{x}_t . In simulation, we first collect human teleoperation demonstrations $\mathcal{D}_{\text{sim}}^{\text{teleop}}$, then use MimicGen [54] to synthesize additional demonstrations $\mathcal{D}_{\text{sim}}^{\text{Mimic}}$ under varied initial object configurations and filter trajectories using the same success predicates as in evaluation; we set $\mathcal{D}_{\text{sim}} = \mathcal{D}_{\text{sim}}^{\text{teleop}} \cup \mathcal{D}_{\text{sim}}^{\text{Mimic}}$. We write $\ell_{\text{flow}}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{A}^*)$ for the standard conditional flow-matching loss on demonstrated action chunk \mathbf{A}^* conditioned on observation \mathbf{x} . The co-training objective is

$$\mathcal{L}_{\text{co}}(\boldsymbol{\theta}) = \alpha \mathbb{E}_{(\mathbf{x}, \mathbf{A}^*) \sim \mathcal{D}_{\text{sim}}}[\ell_{\text{flow}}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{A}^*)] + (1 - \alpha) \mathbb{E}_{(\mathbf{x}, \mathbf{A}^*) \sim \mathcal{D}_{\text{real}}}[\ell_{\text{flow}}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{A}^*)]. \quad (7)$$

The coefficient $\alpha \in [0, 1]$ controls the relative contribution of simulated data. This ratio is not merely a data-balance hyperparameter: effective co-training should preserve task and contact structure across domains without diluting real-only deployment cues [24].

Co-training has two practical effects that it provides the policy with a tactile-conditioned prior grounded in real observations, and it offers sparse-reward simulator RL a viable initialization for

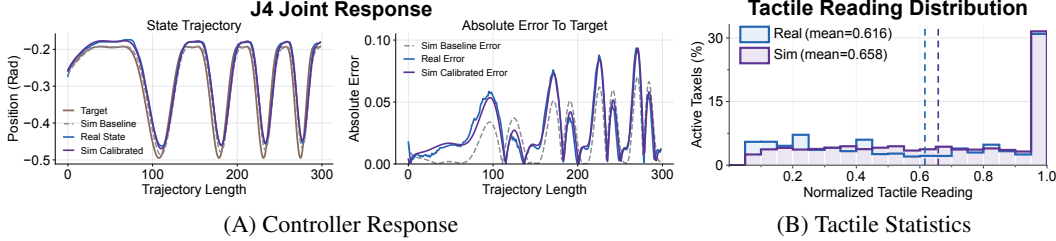


Figure 4: **Controller and tactile alignment.** (A) Held-out J4 joint-response replay comparing the target, real and simulated responses before and after controller SysID. (B) Normalized tactile-reading histogram from matched contact rollouts after tactile calibration.

refining contact behavior. However, this co-training stage remains fixed-dataset imitation that it aligns tactile observations with expert action chunks but provides no on-policy contact rollouts or task-level feedback to improve the policy.

3.4 Post-Training with a Real-Data Anchor

Post-training then optimizes the policy using the on-policy simulation rollouts. We use standard PPO [55] with task-level success and failure predicates as sparse rewards, so the update favors rollouts that complete the task under contact-rich interaction. To prevent simulator-specific drift, we optimize RL jointly with a supervised real-data anchor [56]:

$$\min_{\theta, \omega} \mathcal{L}_{\text{RL}}(\theta, \omega) = \mathcal{L}_{\text{PPO}}(\theta, \omega; E_{\text{sim}}(\psi)) + \beta \mathbb{E}_{(\mathbf{x}, \mathbf{A}^*) \sim \mathcal{D}_{\text{real}}}[\ell_{\text{flow}}(\theta; \mathbf{x}, \mathbf{A}^*)]. \quad (8)$$

Here ω are the critic parameters, \mathcal{L}_{PPO} is the clipped PPO loss, and β weights the real-data behavior cloning loss. This stage collects rollouts from the current tactile-conditioned policy and updates it using sparse task-level rewards. The real-data term anchors the policy to real observations and actions, constraining simulator-specific exploitation [57, 28]. After post-training, we discard the critic, reward, and privileged state and deploy the final policy directly to real robots. Appendix D reports co-training and simulator post-training parameters.

4 Experimental Results

Our experiments address four related questions: (1) whether the simulated controller responses and tactile readings are sufficiently aligned with those of the real to support closed-loop contact learning (Sec. 4.1); (2) whether sim-real co-training provides a strong initialization for simulator RL, and whether tactile feedback further improves the refined policy in simulation (Sec. 4.3); (3) whether policies trained with our method achieve higher success rates on the real (Sec. 4.4); and (4) how the co-training and real-data anchoring weights affect training and real-world deployment (Sec. 4.5).

4.1 Controller and Tactile Alignment

We first validate the two policy-facing interfaces for closed-loop contact learning: controller response and tactile readings. For controller system identification (SysID), we collect short single-joint sweeps on the real robot, replay the same target trajectories in simulation, and identify each joint’s proportional gain K_p , derivative gain K_d , and feed-forward torque T_{ref} to match the real target-tracking errors. For tactile calibration, we replay matched contact trajectories and compare normalized active-taxel readings across domains.

Figure 4 summarizes the alignment of the two policy-facing interfaces. Panel (A) shows a representative held-out J4 replay, selected because gravity-compensation mismatch produces the largest pre- versus post-calibration gap. After controller SysID, the calibrated simulator closely matches the real trajectory and tracking error under the same command. Panel (B) shows similar normalized tactile-reading distributions from matched real and simulated contact rollouts after tactile calibration. Together, these results indicate that the action-execution and contact-observation interfaces are

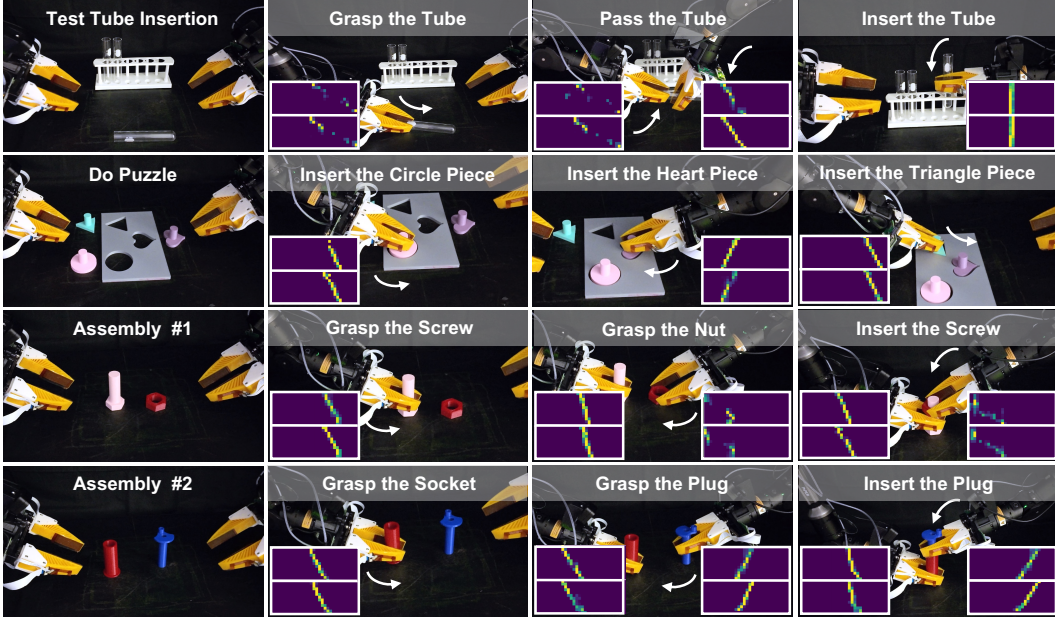


Figure 5: **Real-world policy rollouts.** Representative real-robot executions of our post-trained visuo-tactile VLA policy across four contact-rich bimanual tasks.

sufficiently aligned to support subsequent simulator post-training. Appendices B.1 and B.2 provide additional details.

4.2 Task Settings

We evaluate on the bimanual platform described in Appendix A, using matched camera, action, and tactile interfaces across the real and simulated setups. Figure 3 shows the four contact-rich bimanual tasks and the object-pose randomization used for simulation training; Appendix E provides the full randomization details. For each task, we collect 50 real demonstrations and 20 simulated teleoperation trajectories, expand the simulated set to 200 trajectories using MimicGen, and evaluate with 128 parallel simulation environments and 20 real-world trials per task.

4.3 Sim-Real Co-Training and Simulator RL Post-Training

We evaluate whether sim-real co-training provides a viable initialization for simulator RL and whether tactile feedback further improves the refined policy with RL. Figure 6 shows that RL-trained visuo-tactile policies achieve higher final success than vision-only policies across all tasks. This indicates that tactile histories remain useful during on-policy refinement, rather than serving solely as an additional imitation-learning input.

Table 1 reports RL post-training results for both the co-trained only policy and the same policy evaluated with the exploration noise used at the start of RL. Adding exploration noise lowers the initial average success from 33.0% to 22.5% for vision-only policies and from 40.5% to 29.3% for visuo-tactile policies, but this stochasticity is necessary for sparse-reward exploration. In contrast, direct RL from the base VLA remains at 0.0% across all tasks, indicating that sparse rewards alone cannot bootstrap the task sequence or precise contact corrections.

Sim-real co-training therefore changes the post-training problem from exploration from scratch to policy refinement. After RL, average simulator success increases to 60.5% for vision-only policies and 78.5% for visuo-tactile policies. The tactile gains are especially clear in the *Test Tube Insertion* and *Assembly #2* tasks, where angular error, blockage, and residual misalignment are often visually occluded but remain reflected in tactile histories.

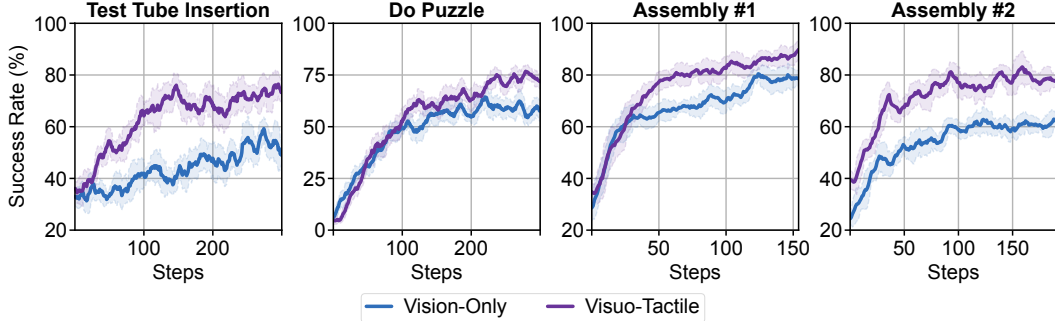


Figure 6: **Tactile feedback improves simulator RL.** Across all tasks, visuo-tactile policies consistently achieve higher success rates than vision-only policies during simulator RL.

Table 1: **Simulation success rates.** Direct RL from the base VLA model fails across all tasks. Sim-real co-training provides a strong initialization and substantially improves the success rate for simulator-based RL. Incorporating tactile feedback further improves policy performance.

| Settings | Simulation Bimanual Setup | | | | | | | |
|---------------------------------|---------------------------|-------------|-------------|-------------|----------------------------|-------------|-------------|-------------|
| | Vision-Only Policy (Sim) | | | | Visuo-Tactile Policy (Sim) | | | |
| | Test Tube Insertion | Do Puzzle | Assembly #1 | Assembly #2 | Test Tube Insertion | Do Puzzle | Assembly #1 | Assembly #2 |
| After Co-Training | 0.42 | 0.12 | 0.39 | 0.39 | 0.41 | 0.16 | 0.47 | 0.58 |
| RL Start with Exploration Noise | 0.35 | 0.05 | 0.25 | 0.25 | 0.36 | 0.04 | 0.35 | 0.42 |
| RL from Base VLA | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| RL with Co-Training | <u>0.50</u> | <u>0.54</u> | <u>0.77</u> | <u>0.61</u> | 0.72 | 0.71 | 0.92 | 0.79 |

4.4 Real-World Evaluation

We then evaluate the extent to which tactile feedback, sim-real co-training, and simulator RL improve policy performance on the real robot.

Figure 5 qualitatively illustrates the sim-to-real transfer of the learned contact behaviors. In these rollouts, the held object or gripper often occludes the target contact region; the post-trained policy uses tactile feedback to correct local contact errors through incremental translations and reorientations. Quantitatively, tactile feedback increases the average success of RL post-trained policies from 50.0% to 72.5% across the four tasks, indicating that the gain comes from contact sensing rather than from simulator RL alone. *Do Puzzle* is the longest-horizon task because it requires all three pieces to be placed successfully. Real-only fine-tuning and sim-real co-training often stop after partial completion, whereas RL post-training reinforces complete trajectories and improves visuo-tactile success to 45%, compared with 15% for real-only fine-tuning and 25% for sim-real co-training, as shown in Table 2. Appendix C provides further analysis of baselines that fail during these occluded contact phases.

4.5 Ablation of Co-Training and Real-Data Anchoring Weights

We further use the *Assembly #2* task to ablate the two hyperparameters that govern sim-to-real co-training and simulator RL: the co-training ratio α sets the relative weight of simulated demonstrations during co-training, while the anchor weight β determines the strength of the real-data anchor during RL. We report the simulator success rate, real-data anchor loss, and real-world deployment success rate across varying parameter settings in Figure 7.

Table 2: **Real-world success rates.** Rows compare training stages; columns compare vision-only and visuo-tactile policies. Tactile feedback and RL post-training together achieve the highest real-world success rates.

| Settings | Real-World Bimanual Setup | | | | | | | |
|-----------------------|---------------------------|-------------|-------------|-------------|-----------------------------|-------------|-------------|-------------|
| | Vision-Only Policy (Real) | | | | Visuo-Tactile Policy (Real) | | | |
| | Test Tube Insertion | Do Puzzle | Assembly #1 | Assembly #2 | Test Tube Insertion | Do Puzzle | Assembly #1 | Assembly #2 |
| Real-Only Fine-Tuning | 0.20 | 0.05 | 0.35 | 0.25 | 0.45 | 0.15 | 0.35 | 0.40 |
| Sim-Real Co-Training | 0.35 | 0.10 | 0.40 | 0.35 | 0.50 | 0.25 | 0.45 | 0.55 |
| RL Post-Training | <u>0.35</u> | <u>0.25</u> | <u>0.80</u> | <u>0.60</u> | 0.70 | 0.45 | 0.95 | 0.80 |

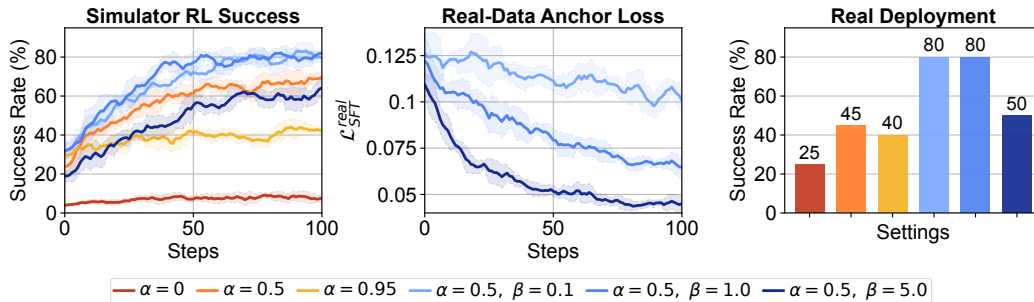


Figure 7: **Ablation of co-training and real-data anchoring.** We vary the co-training ratio α and real-anchor weight β on the *Assembly #2* task and report model performance in terms of simulator success rate (*left*), real-data anchor loss (*middle*), which measures policy deviation from real demonstrations during simulation, and real-world deployment success rate (*right*).

First, we study different choices of α with $\beta = 0$. We observe that lowering α from 0.95 to 0.5 improves the best simulator success rate from 42.9% to 70.3% and increases real-world deployment success from 40% to 45%, indicating that RL benefits from a more balanced sim-to-real co-training initialization. When $\alpha = 0$, simulated demonstrations are removed entirely, so RL starts from a real-only policy in a zero-shot simulation setting. This configuration performs poorly in both domains (14.1% best simulator success and 25% real success), suggesting that the initial simulator rollouts are too far from the policy’s learned interaction distribution for effective post-training.

We then study the effect of β with $\alpha = 0.5$ fixed. Both $\beta = 0.1$ and $\beta = 1.0$ increase real-world deployment success from 45% to 80% and improve the best simulator success to above 92%. This indicates that the real-data anchor provides a useful balance: it prevents simulator RL from drifting away from the real-robot data distribution while still allowing the policy to learn reward-driven contact corrections. Without the anchor, the policy can overfit to simulator-specific contact strategies. In contrast, an overly strong anchor ($\beta = 5.0$) yields the lowest supervised loss but suppresses RL refinement, pulling the policy back toward imitation and reducing both simulator and real-world success.

5 Conclusion

We presented **TacCoRL**, a sim-to-real post-training framework that augments pretrained VLA policies with tactile feedback. By combining sim-real co-training, simulator RL, and a real-data anchor, our framework learns contact-conditioned corrective actions that transfer to the real robot. Across four bimanual contact-rich tasks, our policy improves real-world success over vision-only and imitation-only baselines, highlighting a practical path for integrating tactile feedback into VLA policies via simulation.

6 Limitations

Requirement for real tactile data. Our method reduces the need for large-scale real tactile interaction data, but still uses a small set of real tactile demonstrations as an anchor during post-training. Future work could combine our approach with tactile representation learning and tactile simulation to further reduce the need for real tactile data collection.

Simulation setup effort. The current system uses a task-level digital twin; this is manageable for our setups but still involves manual effort. More automated asset reconstruction, camera alignment, and tactile-system identification would make the pipeline easier to scale.

Hard-to-simulate contacts. Our experiments focus on rigid or near-rigid contact-rich manipulation where the simulator can provide reliable contact predicates and tactile statistics. Tasks involving highly deformable objects or fluids require additional modeling beyond the current setup. Extending our framework to these more complex contact dynamics is an important direction for future work.

References

- [1] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.
- [2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [4] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- [5] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.
- [6] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. In *Conference on Robot Learning*, 2024.
- [7] Z. Zhao, S. Haldar, J. Cui, L. Pinto, and R. Bhirangi. Touch begins where vision ends: Generalizable policies for contact-rich manipulation. *arXiv preprint arXiv:2506.13762*, 2025.
- [8] B. Huang, J. Xu, I. Akinola, W. Yang, B. Sundaralingam, R. O’Flaherty, D. Fox, X. Wang, A. Mousavian, Y.-W. Chao, et al. Vt-refine: Learning bimanual assembly with visuo-tactile feedback via simulation fine-tuning. *arXiv preprint arXiv:2510.14930*, 2025.
- [9] F. Yang, C. Ma, J. Zhang, J. Zhu, W. Yuan, and A. Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022.
- [10] N. Cheng, J. Xu, C. Guan, J. Gao, W. Wang, Y. Li, F. Meng, J. Zhou, B. Fang, and W. Han. Touch100k: A large-scale touch-language-vision dataset for touch-centric multimodal representation. *Information Fusion*, 124:103305, 2025.
- [11] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. 2024. In URL <https://openreview.net/forum>, 2024.
- [12] P. Hao, C. Zhang, D. Li, X. Cao, X. Hao, S. Cui, and S. Wang. Tla: Tactile-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2503.08548*, 2025.
- [13] C. Zhang, P. Hao, X. Cao, X. Hao, S. Cui, and S. Wang. Vtla: Vision-tactile-language-action model with preference learning for insertion manipulation. *Biomimetic Intelligence and Robotics*, page 100333, 2026.
- [14] Z. Cheng, Y. Zhang, W. Zhang, H. Li, K. Wang, L. Song, and H. Zhang. Omnivtla: Vision-tactile-language-action model with semantic-aligned tactile sensing. *arXiv preprint arXiv:2508.08706*, 2025.
- [15] J. Huang, S. Wang, F. Lin, Y. Hu, C. Wen, and Y. Gao. Tactile-vla: unlocking vision-language-action model’s physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.
- [16] J. Bi, K. Y. Ma, C. Hao, M. Z. Shou, and H. Soh. Vla-touch: Enhancing vision-language-action models with dual-level tactile feedback. *arXiv preprint arXiv:2507.17294*, 2025.

- [17] K. Zhang, H. Zhang, Z. Xu, Z. Zhang, M. R. I. Prince, X. Li, X. Han, Y. Zhou, A. Ajoudani, and Y. She. Tacvla: Contact-aware tactile fusion for robust vision-language-action manipulation. *arXiv preprint arXiv:2603.12665*, 2026.
- [18] Z. Zhang, J. Ma, X. Yang, X. Wen, Y. Zhang, B. Li, Y. Qin, J. Liu, C. Zhao, L. Kang, et al. Touchguide: Inference-time steering of visuomotor policies via touch guidance. *arXiv preprint arXiv:2601.20239*, 2026.
- [19] J. Xu, S. Kim, T. Chen, A. R. Garcia, P. Agrawal, W. Matusik, and S. Sueda. Efficient tactile simulation with differentiability for robotic manipulation. In *Conference on Robot Learning*, pages 1488–1498. PMLR, 2023.
- [20] I. Akinola, J. Xu, J. Carius, D. Fox, and Y. Narang. Tacsl: A library for visuotactile sensor simulation and learning. *IEEE Transactions on Robotics*, 2025.
- [21] Y. Li, W. Du, C. Yu, P. Li, Z. Zhao, T. Liu, C. Jiang, Y. Zhu, and S. Huang. Taccel: Scaling up vision-based tactile robotics via high-performance gpu simulation. *Advances in Neural Information Processing Systems*, 38:94577–94604, 2026.
- [22] S. Sha, Y. Wang, B. Huang, A. Loquercio, and Y. Li. Efficient and reliable teleoperation through real-to-sim-to-real shared autonomy. *arXiv preprint arXiv:2603.17016*, 2026.
- [23] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025.
- [24] Y. Lei, M. Liu, A. Maddukuri, Z. Jiang, and Y. Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies. *arXiv preprint arXiv:2604.13645*, 2026.
- [25] S. Tan, K. Dou, Y. Zhao, and P. Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.
- [26] K. Chen, Z. Liu, T. Zhang, Z. Guo, S. Xu, H. Lin, H. Zang, X. Li, Q. Zhang, Z. Yu, et al. π_{RL} : Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025.
- [27] H. Li, Y. Zuo, J. Yu, Y. Zhang, Z. Yang, K. Zhang, X. Zhu, Y. Zhang, T. Chen, G. Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025.
- [28] L. Shi, S. Chen, F. Gao, Y. Chen, K. Chen, T. Zhang, H. Zang, W. Zhang, C. Yu, and Y. Wang. Beyond imitation: Reinforcement learning-based sim-real co-training for vla models. *arXiv preprint arXiv:2602.12628*, 2026.
- [29] X. Zhang, C. Jia, S. Li, D. He, X. Xiong, Z. Sun, H. He, Y. Wu, B. Yu, L. Sun, et al. How rl unlocks the aha moment in geometric interleaved reasoning. *arXiv preprint arXiv:2603.01070*, 2026.
- [30] A. Alspach, K. Hashimoto, N. Kuppaswamy, and R. Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 597–604. IEEE, 2019.
- [31] Z. Zhao, W. Li, Y. Li, T. Liu, B. Li, M. Wang, K. Du, H. Liu, Y. Zhu, Q. Wang, et al. Embedding high-resolution touch across robotic hands enables adaptive human-like grasping. *Nature Machine Intelligence*, 7(6):889–900, 2025.
- [32] H. Choi, Y. Hou, C. Pan, S. Hong, A. Patel, X. Xu, M. R. Cutkosky, and S. Song. In-the-wild compliant manipulation with umi-ft. *arXiv preprint arXiv:2601.09988*, 2026.

- [33] Y. Li, Y. Chen, Z. Zhao, P. Li, T. Liu, S. Huang, and Y. Zhu. Simultaneous tactile-visual perception for learning multimodal robot manipulation. *IEEE Robotics and Automation Letters*, 2026.
- [34] Z. Xu, R. Uppuluri, X. Zhang, C. Fitch, P. G. Crandall, W. Shou, D. Wang, and Y. She. Unit: Data efficient tactile representation with generalization to unseen objects. *IEEE Robotics and Automation Letters*, 2025.
- [35] R. Feng, J. Hu, W. Xia, T. Gao, A. Shen, Y. Sun, B. Fang, and D. Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. *arXiv preprint arXiv:2502.12191*, 2025.
- [36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [37] F. Yang, C. Feng, Z. Chen, H. Park, D. Wang, Y. Dou, Z. Zeng, X. Chen, R. Gangopadhyay, A. Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024.
- [38] K. Gubernatorov, M. Sannikov, I. Mikhailchuk, E. Kuznetsov, M. Artemov, O. F. Ouwatobi, M. Fernando, A. Asanov, Z. Guo, and D. Tsetserukou. Hapticvla: Contact-rich manipulation via vision-language-action model without inference-time tactile sensing. *arXiv preprint arXiv:2603.15257*, 2026.
- [39] C. Xu, J. T. Springenberg, M. Equi, A. Amin, A. Esmail, S. Levine, and L. Ke. Rl token: Bootstrapping online rl with vision-language-action models. *arXiv preprint arXiv:2604.23073*, 2026.
- [40] H. Zang, M. Wei, S. Xu, Y. Wu, Z. Guo, Y. Wang, H. Lin, L. Shi, Y. Xie, Z. Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.
- [41] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, J. DiCarlo, et al. $\pi_{0,6}^*$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.
- [42] H. Zhang, S. Zhang, J. Jin, Q. Zeng, Y. Qiao, H. Lu, and D. Wang. Balancing signal and variance: Adaptive offline rl post-training for vla flow models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18755–18763, 2026.
- [43] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.
- [44] A. Ren, J. Lidard, L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pages 77288–77329, 2025.
- [45] H. Jiang and Z. Yang. Adaptive diffusion policy optimization for robotic manipulation. *arXiv preprint arXiv:2505.08376*, 2025.
- [46] G. Zou, W. Li, H. Wu, Y. Qian, Y. Wang, and H. Wang. D²ppo: Diffusion policy policy optimization with dispersive loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18891–18899, 2026.
- [47] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pages 6023–6029. IEEE, 2019.

- [48] M. Alakuijala, G. Dulac-Arnold, J. Mairal, J. Ponce, and C. Schmid. Residual reinforcement learning from demonstrations. *arXiv preprint arXiv:2106.08050*, 2021.
- [49] K. Fang, W. Liang, Y. Li, J. Zhang, P. Zeng, L. Gao, J. Song, and H. T. Shen. Sim-and-human co-training for data-efficient and generalizable robotic manipulation. *arXiv preprint arXiv:2601.19406*, 2026.
- [50] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *Science Robotics*, 11(113):eaea6201, 2026.
- [51] X. Li, K. Hsu, J. Gu, O. Mees, K. Pertsch, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *8th Annual Conference on Robot Learning*, 2024.
- [52] A. Bronars, Y. Park, and P. Agrawal. Tune to learn: How controller gains shape robot policy learning. *arXiv preprint arXiv:2604.02523*, 2026.
- [53] Y. R. Song, J. Li, R. Fu, D. Murphy, K. Zhou, R. Shiv, Y. Li, H. Xiong, C. E. Owens, Y. Du, et al. Opentouch: Bringing full-hand touch to real-world interaction. *arXiv preprint arXiv:2512.16842*, 2025.
- [54] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.
- [55] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [56] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich. Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments. *arXiv preprint arXiv:1910.04281*, 2019.
- [57] S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.
- [58] B. Huang and Y. Li. Flexitac: A low-cost, open-source, scalable tactile sensing solution for robotic systems. *arXiv preprint arXiv:2604.28156*, 2026.
- [59] N. Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE, 1984.
- [60] B. Katz, J. Di Carlo, and S. Kim. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In *2019 international conference on robotics and automation (ICRA)*, pages 6295–6301. IEEE, 2019.
- [61] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.
- [62] C. Yu, Y. Wang, Z. Guo, H. Lin, S. Xu, H. Zang, Q. Zhang, Y. Wu, C. Zhu, J. Hu, Z. Huang, M. Wei, Y. Xie, K. Yang, B. Dai, Z. Xu, J. Du, X. Wang, X. Fu, L. Shi, Z. Liu, K. Chen, W. Liu, G. Liu, B. Li, J. Yang, Z. Yang, G. Dai, and Y. Wang. RLinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation. *arXiv preprint arXiv:2509.15965*, 2025.
- [63] K. Chen, Z. Liu, T. Zhang, Z. Guo, S. Xu, H. Lin, H. Zang, X. Li, Q. Zhang, Z. Yu, G. Fan, T. Huang, Y. Wang, and C. Yu. π_{RL} : Online RL fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025.

Supplementary Materials

Contents

| | | |
|-----|--|----|
| A | Robot Setup | 13 |
| B | Real-to-Sim-to-Real | 14 |
| B.1 | Controller SysID Details | 14 |
| B.2 | Tactile Signal Alignment Details | 15 |
| B.3 | Camera Calibration Details | 16 |
| C | Baseline Failure Analysis | 17 |
| D | Training Parameters | 18 |
| E | Domain Randomization Parameters | 19 |

A Robot Setup

We conduct all real-world experiments on the bimanual platform in Figure 8. The setup uses two AgileX PiPER 6-DoF robotic arms, a fixed front-view RealSense D415 camera, two wrist-mounted RealSense D405 cameras, and two FlexiTac-V2 tactile pads [58], each pad with a 32×12 taxel array, mounted on the opposing gripper contact surfaces.

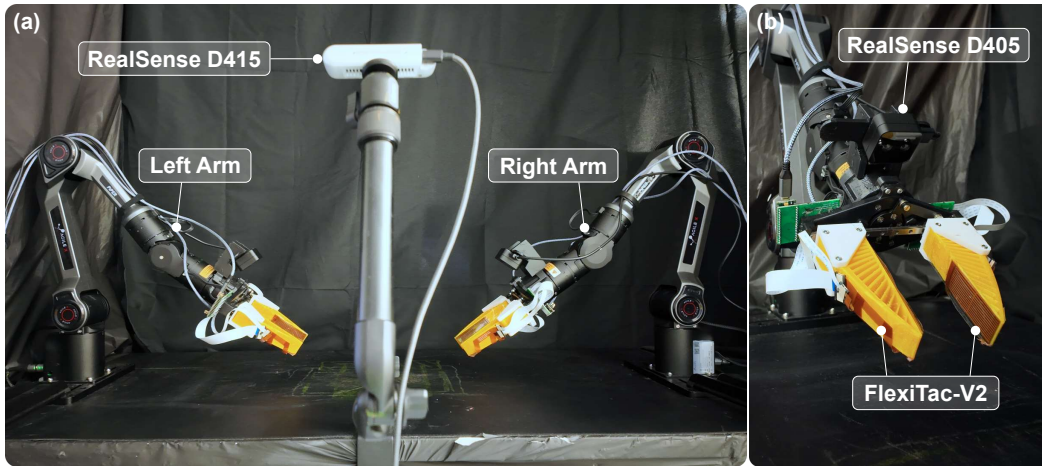


Figure 8: **Real-world robot setup.** (a) Bimanual platform with two AgileX PiPER 6-DoF robotic arms and a fixed front-view RealSense D415 camera. (b) End-effector close-up with wrist-mounted RealSense D405 cameras and two FlexiTac-V2 tactile pads on the gripper contact surfaces.

B Real-to-Sim-to-Real

B.1 Controller SysID Details

For each joint j , we record a 10-second frequency sweep while holding the other joints fixed, then replay the same target sequence in simulation. We tune the simulated MIT-style impedance controller [59, 60] by optimizing $\eta_j = [\log K_p, \log K_d, T_{\text{ref},j}]$ with SPSA [61], using the loss

$$\mathcal{L}_j = \sum_t |e_{t,j}^{\text{sim}} - e_{t,j}^{\text{real}}| + 0.1 \sum_t |\Delta e_{t,j}^{\text{sim}} - \Delta e_{t,j}^{\text{real}}|, \quad e_{t,j}^\rho = q_{t,j}^\rho - q_{t,j}^{\text{tar}}, \quad \rho \in \{\text{real}, \text{sim}\}. \quad (9)$$

The first term matches target-tracking bias, while the finite-difference term constrains response speed and damping. The search bounds are $K_p \in [10, 10^5] \text{ N}\cdot\text{m}\cdot\text{rad}^{-1}$, $K_d \in [1, 10^3] \text{ N}\cdot\text{m}\cdot\text{s}\cdot\text{rad}^{-1}$, and $T_{\text{ref}} \in [-8, 8] \text{ N}\cdot\text{m}$. The reported runs use SPSA with $a_{\text{SPSA}} = 0.15$, $c_{\text{SPSA}} = 0.08$, $A_{\text{SPSA}} = 20$, $\alpha_{\text{SPSA}} = 0.602$, $\gamma_{\text{SPSA}} = 0.101$, gradient clipping at 5, and early stopping patience of 20 iterations. Please refer to Figure 9 for the full PD calibration results.

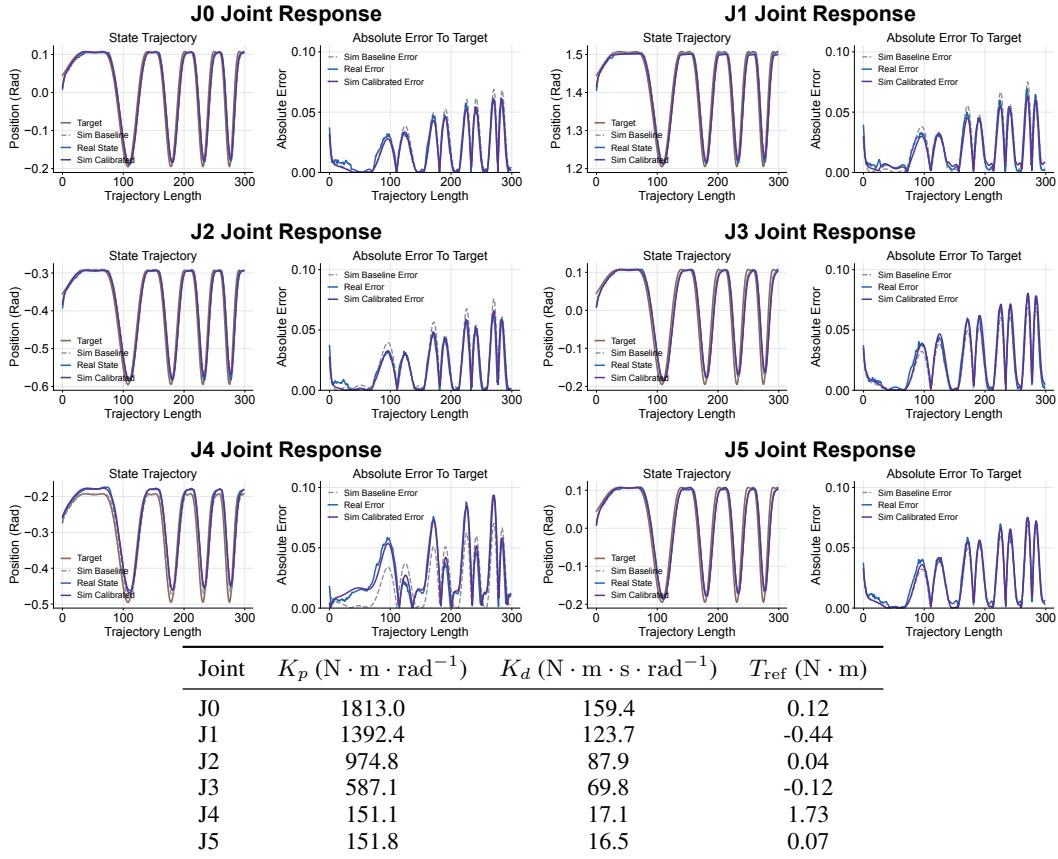


Figure 9: **Per-joint controller SysID.** Each panel replays the same real single-joint sweep in simulation. The reference simulator uses $K_p = 500 \text{ N}\cdot\text{m}\cdot\text{rad}^{-1}$, $K_d = 50 \text{ N}\cdot\text{m}\cdot\text{s}\cdot\text{rad}^{-1}$, and $T_{\text{ref}} = 0 \text{ N}\cdot\text{m}$; the calibrated simulator uses the identified parameters listed below the response plots. SysID reduces lag, overshoot, and steady-state bias across joints.

B.2 Tactile Signal Alignment Details

Let \mathbf{o}_t^r be the tactile arrays flattened into K taxels with scalar readings $f_{t,k}$. Following spring-damper tactile simulators [20, 8], each taxel queries the signed distance to the contacting object and maps penetration depth $\delta_{t,k} \geq 0$ and penetration-rate $\dot{\delta}_{t,k}$ to a normal contact reading:

$$f_{t,k}^{\text{sim}} = \left[k_n \delta_{t,k} + k_d \dot{\delta}_{t,k} \right]_+, \quad [z]_+ = \max(z, 0), \quad (10)$$

where k_n and k_d are normal stiffness and damping. Because the policy consumes scalar taxel histories, we align this normal channel rather than reconstructing shear or full force fields.

Alignment calibrates (k_n, k_d) and the policy-facing normalization. We estimate λ_{noise} from no-contact real readings to suppress background noise and define active-taxel masks, then tune (k_n, k_d) on matched real-to-sim contact replays to align active-taxel masks, active-taxel counts, centroids, histograms, and temporal differences. The same thresholding and normalization are applied in both domains; Figure 10 shows the resulting side-by-side *Assembly #1* and *Assembly #2* replays.



Figure 10: **Qualitative tactile signal alignment.** Side-by-side real-to-sim replay trajectories for Assembly #1 and #2. Each pair shows synchronized real and simulated frames with tactile maps, highlighting matched contact location and evolution.

B.3 Camera Calibration Details

Camera extrinsics are calibrated offline by replaying a synchronized real episode in simulation and optimizing small pose offsets around the nominal camera mounts. Figure 11 provides qualitative evidence of the resulting visual alignment across the initial, grasp, and insertion stages.



Figure 11: **Camera calibration.** Simulation-to-real alignment after camera extrinsic calibration. Columns show the fixed front camera, left wrist camera, and right wrist camera. The top two rows compare rendered simulation views with synchronized real views, and the bottom rows overlay the two domains at the initial state, grasp phase, and insertion phase.

C Baseline Failure Analysis

Figure 12 shows one representative failure from each task; the full failure distribution is broader, but these examples share a post-contact ambiguity. After first contact, the rack rim, puzzle-hole boundary, or assembly mating interface is hidden by the gripper or held part, so success depends on converting local contact cues into corrective motion rather than continuing the nominal trajectory.

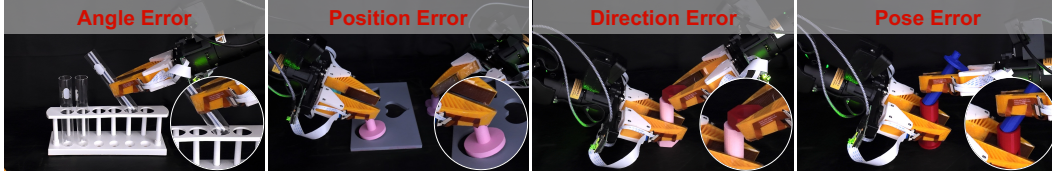


Figure 12: **Representative baseline failure cases.** Each failure occurs after the gripper or held object occludes the local contact region. These cases expose missing corrective behavior: angle correction after rack-rim contact, lateral search for the puzzle opening, axis realignment in Assembly #1, and center-of-mass support before release in Assembly #2.

Test Tube Insertion: A small angular misalignment between the tube and rack opening causes the tube to catch on the rim. Once the gripper and tube hide the rack-rim contact line, vision cannot verify centering. The imitation-only tactile baseline detects contact but keeps inserting instead of reorienting, so the tube gets stuck.

Do Puzzle: A small lateral offset can leave the piece outside the target hole even when its orientation is roughly correct. The piece and gripper partially hide the hole boundary, and object symmetry makes the residual offset difficult to infer from vision alone. The imitation-only tactile baseline repeatedly presses at the same wrong location rather than translating to search, pushing the rollout into more out-of-distribution failure states.

Assembly #1: An incorrect insertion direction after grasping can make the parts contact while the relative motion is not aligned with the true insertion axis. Once the held part and gripper hide the mating edge, the vision-only baseline continues the wrong approach, causing sliding or wedging.

Assembly #2: Coupled translation and orientation error can leave the plug misaligned with the socket. Once the plug and gripper occlude the socket opening, the vision-only baseline may insert at an incorrect angle while the plug's center of mass remains outside the socket. After release, the plug is unsupported and falls out.

D Training Parameters

We list training parameters for Secs. 3.3 and 3.4. Table 3 covers fixed-dataset sim-real co-training for a tactile-augmented $\pi_{0.5}$ VLA model [4]. For simulator post-training, the flow-policy RL implementation follows RLinf and π_{RL} [62, 63]; Table 4 records the rollout, critic, PPO, and optimization settings.

Table 3: **Sim-real co-training parameter values.** Fixed-dataset sim-real co-training hyperparameters for a tactile-augmented $\pi_{0.5}$ VLA model [4].

| Group | Symbol | Meaning | Value |
|----------|---------------------|----------------------------------|---|
| Data | N_{real} | real demonstration episodes | 50 |
| | N_{sim} | simulated demonstration episodes | 200 |
| | α | simulation loss weight | 0.5 |
| Tactile | L | tactile history length | 10 |
| | K | policy-facing taxels | $4 \times 32 \times 12 = 1536$ |
| | E_{τ} | tactile encoder | CNN2 temporal token |
| | S_{τ} | tactile injection site | prefix + suffix tokens |
| | M_{pre} | prefix tactile tokens | 8 |
| | M_{suf} | suffix tactile tokens | 8 |
| | λ_f | contact gate threshold | 0.4 |
| | m | active-count threshold | 1 |
| Model | H | action horizon | 10 |
| | d_a | model output action dimension | 14 |
| | θ_0 | initialization | $\pi_{0.5}$ base VLA |
| Training | σ_v | image noise standard deviation | 0.03 |
| | B | global batch size | 192 |
| | T_{warm} | warmup steps | 100 |
| | η_{max} | peak learning rate | 2.5×10^{-5} |
| | η_{min} | final learning rate | 5×10^{-6} |
| | T_{co} | co-training steps | Assembly #1/#2: 1000 Do Puzzle/Test Tube Insertion: 3000 |
| | | \mathcal{O} | optimizer |

Table 4: **Post-training parameter values.** Simulator post-training hyperparameters for Sec. 3.4, including flow-policy RL, rollout, critic, PPO, and optimization settings.

| Group | Symbol | Meaning | Value |
|--------------|-----------------------------------|---------------------------------|-----------------------|
| Anchor data | $N_{\text{real}}^{\text{anchor}}$ | real-anchor episodes | 200 |
| | B_{real} | real-anchor batch size | 32 |
| | β | real-anchor loss weight | 1.0 |
| Rollout | N_{env} | parallel simulator environments | 128 |
| | C | rollout chunk steps | 36 |
| Flow policy | $\mathcal{M}_{\text{flow}}$ | flow-policy RL formulation | Flow-SDE |
| | σ_{SDE} | Flow-SDE noise level | 0.5 |
| | K_{denoise} | denoising steps | 4 |
| | ρ_{chunk} | reward and log-prob aggregation | chunk level |
| PPO | \mathcal{A} | advantage estimator | GAE |
| | V_{ϕ} | critic design | mean-token value head |
| | γ | discount factor | 0.99 |
| | λ_{GAE} | GAE trace parameter | 0.95 |
| | ϵ_{clip} | PPO clip ratio | 0.2 |
| | E_{update} | PPO update epochs | 1 |
| Optimization | B_{RL} | RL global batch size | 512 |
| | η_{π} | actor learning rate | 5×10^{-6} |
| | η_V | value-head learning rate | 1×10^{-4} |

E Domain Randomization Parameters

Table 5 summarizes the domain randomization used in simulation for each task. We include image-space augmentation and task-level reset randomization for object pose and physical properties.

Table 5: **Domain randomization parameter values.** Simulation domain-randomization values used across the tasks in Sec. 4.2.

| Task | Parameter | Probability | Distribution |
|---------------------|---------------------------|-------------|--|
| All tasks | camera crop-resize | 1.0 | $f_{\text{crop}} = 0.95$ |
| | camera rotation | 1.0 | $\Delta\theta \sim \mathcal{U}([-5^\circ, 5^\circ])$ |
| | camera brightness jitter | 1.0 | $j_b \sim \mathcal{U}([0.7, 1.3])$ |
| | camera contrast jitter | 1.0 | $j_c \sim \mathcal{U}([0.6, 1.4])$ |
| | camera saturation jitter | 1.0 | $j_s \sim \mathcal{U}([0.5, 1.5])$ |
| | object friction | 1.0 | $\mu \sim \mathcal{U}([0.3, 0.7])$ |
| Assembly #1/#2 | plug/socket base position | 1.0 | $\Delta\mathbf{p} \sim \mathcal{U}([-0.03, 0.03] \text{ m})$ |
| | plug/socket mass | 1.0 | $\Delta m \sim \mathcal{U}([-0.03, 0.03] \text{ kg})$ |
| Do Puzzle | tray planar position | 1.0 | $\Delta x, \Delta y \sim \mathcal{U}([-0.005, 0.005] \text{ m})$ |
| | tray yaw | 1.0 | $\Delta\psi \sim \mathcal{U}([-5^\circ, 5^\circ])$ |
| | piece planar position | 1.0 | $\Delta x, \Delta y \sim \mathcal{U}([-0.01, 0.01] \text{ m})$ |
| | piece yaw | 1.0 | $\Delta\psi \sim \mathcal{U}([-10^\circ, 10^\circ])$ |
| | circle reset support | 1.0 | $\sim \mathcal{U}([-0.03, 0.03]^2 \text{ m})$ |
| | heart reset support | 1.0 | $\sim \mathcal{U}([-0.03, 0.03] \times [-0.027, 0.027] \text{ m})$ |
| | triangle reset support | 1.0 | $\sim \mathcal{U}([-0.03, 0.03] \times [-0.027, 0.027] \text{ m})$ |
| Test Tube Insertion | tube planar position | 1.0 | $\Delta x, \Delta y \sim \mathcal{U}([-0.02, 0.02] \text{ m})$ |
| | tube yaw | 1.0 | $\Delta\psi \sim \mathcal{U}([-20^\circ, 20^\circ])$ |
| | holder planar position | 1.0 | $\Delta x, \Delta y \sim \mathcal{U}([-0.005, 0.005] \text{ m})$ |