Appendix

A Implementation Details

A.1 Model Fine-Tuning

All Vision-Language-Action (VLA) policies in this work are derived from a pre-trained base model π_0. The fine-tuning process is conducted as follows:

- **Parameter Update:** For both the hand-only and the arm-hand VLA policies, we perform full-parameter fine-tuning on π_0, with the exception of the visual encoder, which remains frozen.
- **Task Instruction:** The specific language instruction provided to the model is tailored to the task:
  - Instruction for the **hand-only VLA** policy: pick up the object on the table and place it elsewhere.
  - Instruction for the **arm-hand VLA** policy: pick up the object on the table and place it in the box.

A.2 Hardware Platforms and Real-World Deployment

Figure 17 Cross-embodiment validation using two different robotic hands for both the Shared Autonomy framework and the training of end-to-end VLA models: (a) XHAND1 hand [2], and (b) RY-H2 hand [1].

As the research community embraces a range of different designs of hardware, we validated the applicability of our proposed framework using two different dexterous robotic hands which are quite representative (one is fully actuated and the other is underactuated): (1) `Xhand`, a high-performance 12-DoF dexterous robotic hand with fingertip tactile sensing, as the main hardware; (2) RY-H2, a fast 11-DoF dexterous hand (6 active DoF and 5 under-actuated DoF) with joint current sensing featured by a quick open-close cycle of 0.4s.

The `Xhand` provides a high-bandwidth motor control interface for learning-based control, joint-level state data, and high-resolution tactile feedback [2]. Its five fingertips are equipped with 270° encirclement of tactile array sensors, providing 120 channels in total for the 3-dimensional force data per fingertip, providing tactile perception of contact geometry and distributed forces. The motors offer multiple control modes, including position, force, and hybrid force-position control, running at a control frequency of 83 Hz over an EtherCAT. With a fingertip grip force of 15 N and a maximum grip force of 80 N, the hand can perform both precise and powerful grasps. Its design of back-drivable actuation and a lifetime of 1000000 grasp cycles, makes it suitable for the extensive trial-and-error required in data collection and testing trials of VLA policies.

The RY-H2 hand is a five-finger under-actuated dexterous hand featured by high-speed grasping [1]. Totaling 11 joints, with 6 active and 5 passive degrees of freedom, it is actuated by high-power-density brushless DC motors. This design enables a rapid open-close cycle time of 0.4s and a high maximum grip force of 140N. With a lightweight of 0.6kg, it is suitable for dynamic tasks requiring both speed and power, and secondary algorithmic development for industrial and research applications.

During the evaluation phase on these physical robot hardware, the same setting was configured:

- **Model Checkpoint:** All VLA models evaluated in the main experiments and the appendix are the checkpoints saved at 80,000 training steps.
- **Control Frequency:** The robot arm and hand are controlled by the policy at **30 Hz** control frequency.

A.3 Network Architecture Specifications

The Arm-Hand Feature Enhancement module extends the base architecture with dedicated components for limb-specific feature extraction. The arm encoder $\mathcal{E}_{\text{arm}}$ and the hand encoder $\mathcal{E}_{\text{hand}}$ are both implemented as a two-layer MLP. Each encoder takes the shared representation $z_t^{\text{share}} \in \mathbb{R}^{d_s}$ as input and produces limb-specific features of reduced dimensionality $z_t^{\text{arm}} \in \mathbb{R}^{d_s/2}$ and $z_t^{\text{hand}} \in \mathbb{R}^{d_s/2}$ through successive linear transformations separated by the Mish activation functions.

The auxiliary prediction heads $\mathcal{H}_{\text{arm}}$ and $\mathcal{H}_{\text{hand}}$ are implemented as single linear layers that map the limb-specific features to action predictions of a fixed max dimension, with selective supervision applied only to the indices corresponding to each limb's actual degrees of freedom.

B Additional Results

B.1 Effectiveness of Shared Autonomy Data Collection

Table 4 Data collection efficiency and training/deploying expenditure of shared autonomy vs full teleoperation.

Methods	Main Collection	Corrective Collection	Fine-Tuning Time	Deploying Time
Shared Autonomy	110/hour/person	100/hour/person	4 hours (on 4 GPUs)	10–15 minutes (20 trials)
Full Teleoperation	90/hour/person	80/hour/person	NA	NA

Our Shared Autonomy framework demonstrates a clear advantage in data collection efficiency. As shown in Table 4, it allows a single operator to collect 110 trajectories per hour for the main dataset, compared to 90 with full teleoperation. This 25% increase in collection rate, sustained during corrective data collection (100 vs. 80 trajectories/hour), directly translates to faster policy improvement cycles and validates the framework's effectiveness.

This high-quality data allows for efficient policy refinement: a fine-tuning run (20k steps on 4 GPUs) completes in 4 hours, and deploying the refined policy for 20 evaluation trials takes only 10–15 minutes. This end-to-end efficiency shows the feasibility of our approach for rapid policy training and iteration. For skilled operators, collecting more than 100 demos per hour per person is easily and readily achievable, and most domain-specific cases usually require around 50 demos only, which enables a development-to-deployment cycle of one day only.

B.2 Additional Results of End-to-End Adaptive Arm-Hand Grasping

This section presents additional qualitative results, and Fig. 18 to Fig. 19 demonstrate the robustness and generalization capability of our end-to-end VLA policy in additional grasping scenarios.

Figure 18 Grasping performance across different object orientations.

Figure 19 Grasping robustness tested at various spatial locations within the workspace.

B.2.1 Effectiveness of Tactile Sensing in π_{uni}

To evaluate the impact of tactile sensing on the full arm-hand policy, we conducted an additional experiment under the same conditions as our main `Xhand` evaluation (20×20 cm workspace, 10 objects, 10 trials each). The tactile representation and integration method followed the same approach as in the hand-only DexGrasp-VLA policy. Despite the tactile sensing provides significant benefits to the hand-only policy π_{hand}, we found that incorporating these same tactile features into the unified arm-hand policy π_{uni} does not consistently improve performance. In other words, tactile sensing is more effective when used in combination with local visual sensing for the hand-only DexGrasp VLA policy, but the direct incorporation of tactile sensing in the unified arm-hand policy π_{uni} does *not* yield positive results, at least from this initial study. Specifically, the enhanced policy with tactile input ($\pi_{\text{uni-enhance-tac}}$) achieved a success rate of 82%, compared to 95% for the visual-proprioceptive only policy ($\pi_{\text{uni-enhance}}$) trained by datasets collected by shared autonomy.

This performance degradation is likely due to the different functional roles for controlling the arm and the hand respectively. To the best of our knowledge, we hypothesize that the arm primarily executes reaching motions that rely more on visual and proprioceptive feedback for spatial motions, while tactile signals are most relevant for fine-grained grasping and in-hand manipulation. Uniformly incorporating tactile input throughout the entire arm-hand trajectory may introduce irrelevant information like "noises" during arm movement phases, particularly from incidental environmental contacts (e.g., table collisions or unintended fingertip brushing) that occur during reaching. These transient and often misleading tactile signals appear to interfere with the policy's ability to maintain robust arm-centric coordination.

The above results suggest that future work should explore more structured tactile integration strategies rather than uniform feature fusion throughout the entire motion. Promising directions include selective sensor gating mechanisms that activate tactile processing only during grasping phases, or attention-based architectures that learn to dynamically weight tactile input based on the current task phase. Such approaches could preserve the benefits of tactile sensing for manipulation, while avoiding the performance degradation observed during arm movements.

B.2.2 Additional Results of Corrective Control for Refining a VLA Policy

This appendix extends the experimental validation of our corrective framework beyond the pick-and-place tasks presented in the main text. Together with the shared autonomy approach used for grasping tasks in the main experiments, the additional studies here, which employed teleoperation for long-horizon tasks and motion planning for industrial assembly, provide comprehensive evidence for the generality of our corrective mechanism across diverse tasks and data collection methodologies.

B.3 Long-Horizon Manipulation with Robotic Gripper

B.3.1 Long-Horizon Tasks Learned from Teleoperation

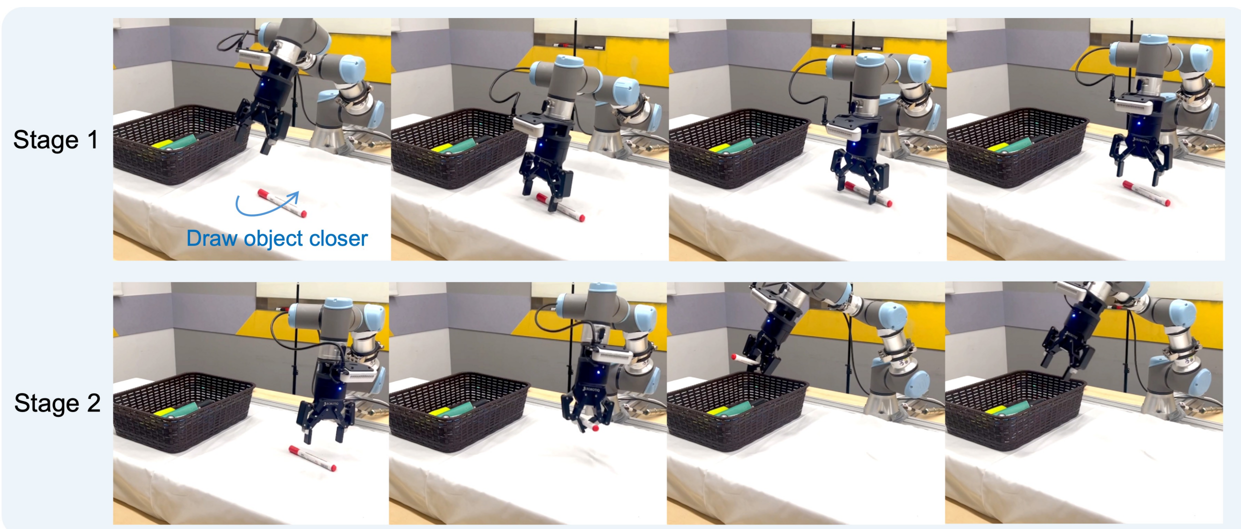

Figure 20 Task 1. Sequential object manipulation via multi-stage physical interactions with a distant object. The task involves a two-stage strategy to manipulate a pen from a distant, non-graspable location (at the arm's far reach near singularity if performing the usual grasp) to a nearby location suitable for grasping. Stage 1: the robot sweeps, reorients, and draws the object closer to its base. Stage 2: the robot executes a final grasp and placement.

Building upon the corrective framework validated in the main text's using dexterous hands, we further designed three sequential long-horizon tasks using parallel jaw grippers to evaluate the framework's efficacy under different hardware and task settings. These three tasks include:

Task 1. Pen Relocation and Placement (Fig. 20): This task requires the robot to first sweep a distant slender pen and bring it to a closer range, then to reorient the gripper to an appropriate angle, grasp and place the pen. Common failure modes include inadequate sweeping force and incorrect gripper orientation during the pre-grasp phase. In this scenario, the robot needs to first re-configure the object and change it from a non-graspable state into a graspable pose, which requires the design of RL policies previously with handcrafted reward design [45] and now can be learned through a VLA-based imitation by providing demonstrations for learning such behaviors.

Task 2. Pill Box Packing (Fig. 21): In this multi-stage task, the robot must sequentially grasp a small medication box, put it into a packaging bag, securely close the bag, and finally transport the entire package to a designated location. Failures typically occur during the delicate bag-closing phase and when handling the combined object during transport.

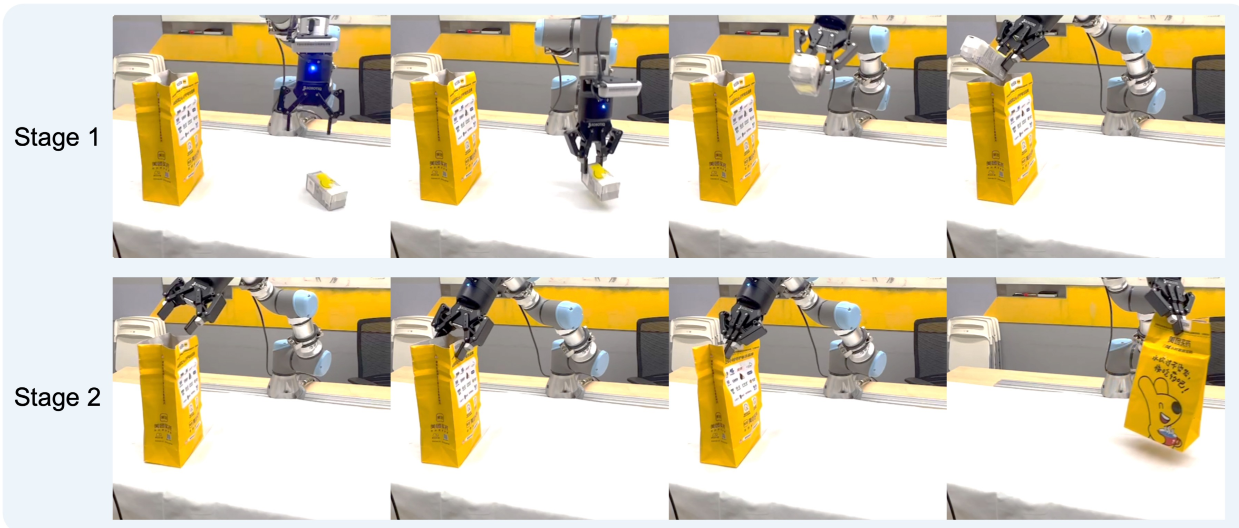

Figure 21 Task 2. Sequential multi-stage packing showing key actions. Stage 1: placing the medication box into the packaging bag. Stage 2: execution of closing the bag and transporting the closed package, showing the capability of operating both rigid and deformable objects.

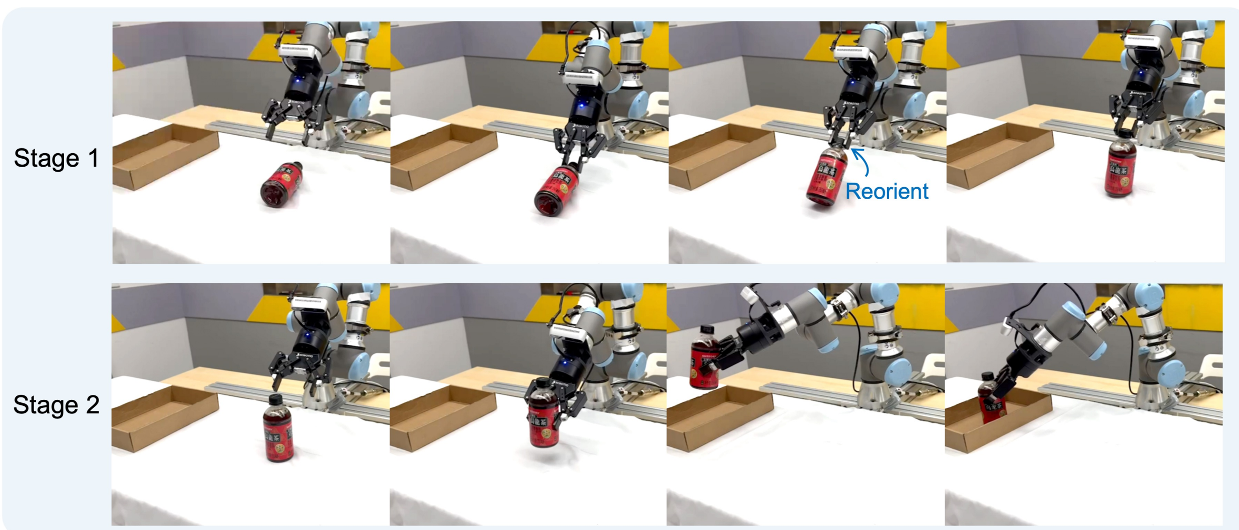

Figure 22 Task 3. Underactuated bottle uprighting by exploiting low torsional friction. Stage 1: grasping the cap and lifting the bottle to induce passive uprighting via low torsional friction around the cap. Stage 2: subsequent pick-and-place from a re-configured graspable pose to the target box.

Task 3. Bottle Uprighting and Placement (Fig. 22): This challenging task begins with re-configuring the state of the object – grasping the cap of a horizontally positioned water bottle, followed by carefully lifting and uprighting the bottle into a vertical orientation – and then completes with a standard pick-and-place operation. The uprighting process presents particular difficulties in maintaining a stable grip, taking advantage of the low torsional friction for passive rotation of the bottle during the execution of the lifting trajectory.

Following our established framework, we first trained a base policy π_{base} on initial demonstrations. Then, we collected corrective trajectories via human teleoperation for those occurred failures across these diverse task scenarios. After fine-tuning, the resulting policy π_{corr} achieved significantly higher success rates (see Table 5), demonstrating that the corrective mechanism (as previously shown effective with shared autonomy) also works well for the hardware system using teleoperation in complex multi-step tasks with parallel jaw grippers.

Table 5 Success rates of long-horizon tasks via corrective teleoperation.

Tasks	**Task 1** (Fig. 20)	**Task 2** (Fig. 21)	**Task 3** (Fig. 22)
Success Rates	65%	90%	70%

B.3.2 Learning Industrial Assembly Task - Data Collection through Motion Planning

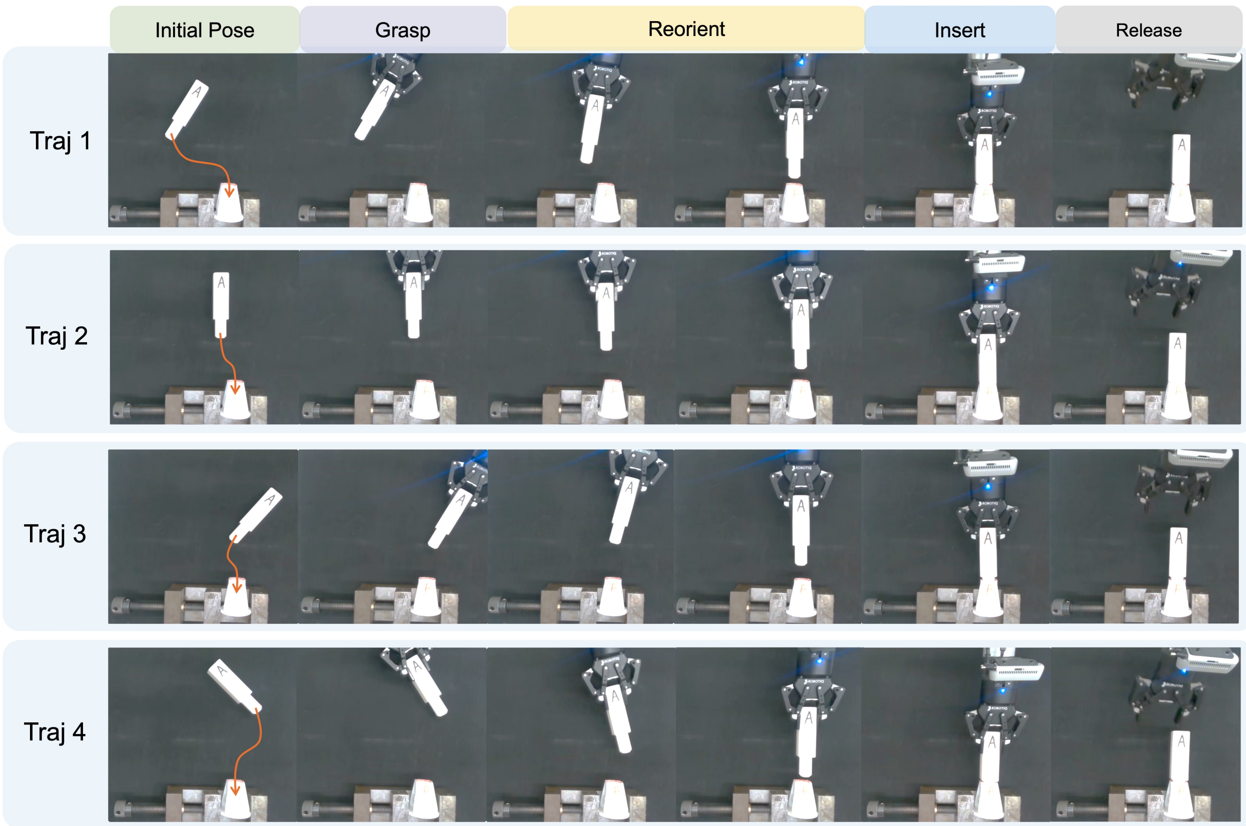

Figure 23 Peg-in-hole assembly task showing multiple stages (grasp, reorient, insert, release) under four different initial configurations.

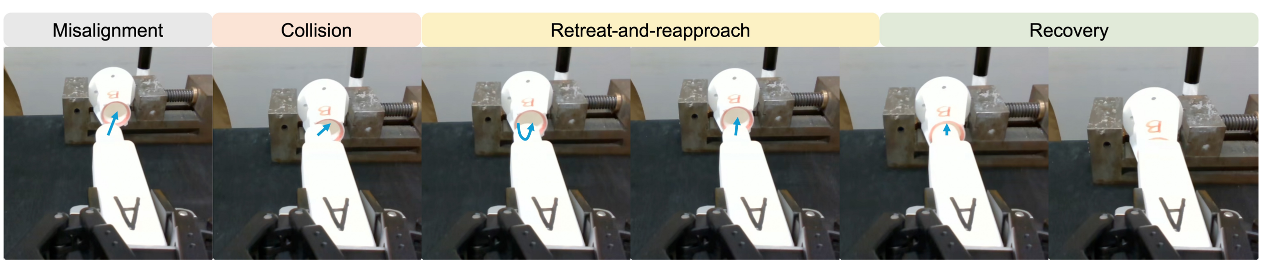

Figure 24 Time-elapsed frames of error-recovery during peg-in-hole assembly, showing stages of misalignment, collision, retreat-and-reapproach, and failure recovery with successful insertion.

Complementing the human-in-the-loop approaches (such as the shared autonomy in the main text and the teleoperation in long-horizon tasks in the Appendix), we investigated how *automated motion planning* could generate effective failure-correction and serve as a useful source of data. We applied our framework to a peg-in-hole assembly task – a canonical industrial task that requires spatial accuracy beyond typical pick-and-place. Following the same experimental protocol, we began with 100 initial demonstrations to train π_{base} using motion planning, where the representative cases from data collection are shown in Fig. 23.

When the policy failed on precision-critical cases, we employed **automated motion planning** to generate corrective trajectories (Fig. 24), instead of human intervention. The motion planner produced recovery behaviors, including fine adjustments and retreat-and-reapproach motions. Reported as in Table 6, after incorporating 20 additional recovery trajectories and fine-tuning, π_{corr} achieved a 90% success rate: **20% absolute improvement** over π_{base}. This result confirms that the corrective mechanism works effectively even with fully automated correction and its generated data sources, extending its applicability beyond human-guided interventions.

Table 6 Success rates of the task of peg-in-hole.

Methods	π_{base}	π_{corr}
Peg-in-Hole	70%	90%

B.4 Discussion and Summary

The collective evidence from all experiments – spanning pick-and-place tasks with shared autonomy (main text), long-horizon tasks with teleoperation, and industrial assembly with motion planning – consistently demonstrates the effectiveness of our corrective framework across task types and data collection approaches. Each study followed the same fundamental principle: initial policy training, failure identification, corrective data collection, and policy re-training/refinement, while employing different failure-correction approaches and data sources tailored to the task-specific requirements. This iterative approach, despite relying on manual intervention and human involvement, represents a preliminary implementation of a closed-loop data flywheel that continuously retrains our VLA models with real-world interactions and the correspondingly generated real-robot data.

This progression from human-guided corrections (e.g., shared autonomy, teleoperation) to fully automated solutions (e.g., motion planning) highlights the framework's core versatility, which can be tailored to different functional components for a wide range of tasks. The shared autonomy approach balances human expertise with automated assistance for efficient grasping corrections; teleoperation provides full human control for complex multi-step tasks; while motion planning offers a fully automated solution for structured industrial environments.

Most importantly, all three approaches yield significant performance improvements, confirming that the corrective mechanism itself is the key driver of policy refinement and enhancement, while any specific data collection method merely serves as a means to this end. This inherent versatility makes our framework applicable across a broad spectrum of robotic learning scenarios, from human-centric environments to structured industrial settings. By providing an effective means for continuous improvement, our work paves the way for more capable and robust general-purpose VLA policies, ultimately expanding the reach of advanced autonomous systems in the real world.

References

[1] RY-H2. URL http://www.ruiyanrobot.com/product/hand/35.

[2] XHAND1. URL https://www.robotera.com/en/goods1/4.html.

[3] Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. Vla-touch: Enhancing vision-language-action models with dual-level tactile feedback. arXiv:2507.17294, 2025.

[4] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. arXiv:2503.14734, 2025.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. arXiv.2410.24164.

[6] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. arXiv:2503.06669, 2025.

[7] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[8] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In IEEE International Symposium on System Integrations (SII), 2019.

[9] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, 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.

[10] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(5):2804–2818, 2023.

[11] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. arXiv:2407.01512, 2024.

[12] Zhengxue Cheng, Yiqian Zhang, Wenkang Zhang, Haoyu Li, Keyu Wang, Li Song, and Hengdi Zhang. Omnivtla: Vision-tactile-language-action model with semantic-aligned tactile sensing. arXiv:2508.08706, 2025.

[13] Aidan Curtis, Nishanth Kumar, Jing Cao, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Trust the PRoC3S: Solving Long-Horizon Robotics Problems with LLMs and Constraint Satisfaction. In Conference on Robot Learning, pages 1362–1383. PMLR, 2025.

[14] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. arXiv:2505.03233, 2025.

[15] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. arXiv:2407.03162, 2024.

[16] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. arXiv:2406.18915, 2024.

[17] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, et al. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning. arXiv preprint arXiv:2504.18904, 2025.

[18] Johanna Hansen, Francois Hogan, Dmitriy Rivkin, David Meger, Michael Jenkin, and Gregory Dudek. Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning. In IEEE International Conference on Robotics and Automation, pages 8298–8304, 2022.

[19] Wenbin Hu, Bidan Huang, Wang Wei Lee, Sicheng Yang, Yu Zheng, and Zhibin Li. Dexterous in-hand manipulation of slender cylindrical objects through deep reinforcement learning with tactile sensing. Robotics and Autonomous Systems, 186:104904, 2025.

[20] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. arXiv:2410.24091, 2024.

[21] 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:2507.09160, 2025.

[22] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. arXiv:2409.01652, 2024.

[23] 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, 2025. URL https://arxiv. org/abs/2504.16054, 1(2):3.

[24] 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. arXiv:2403.12945, 2024.

[25] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. arXiv:2406.09246, 2024.

[26] Nishanth Kumar, William Shen, Fabio Ramos, Dieter Fox, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Caelan Reed Garrett. Open-world task and motion planning via vision-language model inferred constraints. arXiv:2411.08253, 2024.

[27] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. IEEE Transactions on Robotics, 36(3):582–596, 2020.

[28] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. arXiv:2404.16823, 2024.

[29] Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. arXiv:2502.20396, 2025.

[30] Fangchen Liu, Chuanyu Li, Yihua Qin, Ankit Shaw, Jing Xu, Pieter Abbeel, and Rui Chen. Vitamin: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface. arXiv:2504.06156, 2025.

[31] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. arXiv:2503.10631, 2025.

[32] Lilu Liu, Jingyu Zhang, Fei Wang, Jiyu Yu, Yuxiang Cui, Zhibin Li, Jian Hu, Rong Xiong, Haojian Lu, and Yue Wang. AI search, physician removal: Bronchoscopy robot bridges collaboration in foreign body aspiration. Science Robotics, 10(104):eadt5338, 2025.

[33] Qingtao Liu, Yu Cui, Zhengnan Sun, Gaofeng Li, Jiming Chen, and Qi Ye. Vtdexmanip: A dataset and benchmark for visual-tactile pretraining and dexterous manipulation with reinforcement learning. In The Thirteenth International Conference on Learning Representations.

[34] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: vision-language-action pretraining from large-scale human videos. arXiv preprint arXiv:2507.15597, 2025.

[35] 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.

[36] Xiaofeng Mao, Gabriele Giudici, Claudio Coppola, Kaspar Althoefer, Ildar Farkhatdinov, Zhibin Li, and Lorenzo Jamone. DexSkills: Skill segmentation using haptic data for learning autonomous long-horizon robotic manipulation tasks. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5104–5111. IEEE, 2024.

[37] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 27649–27660, 2025.

[38] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. arXiv preprint arXiv:2406.02523, 2024.

[39] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In 2024 IEEE International Conference on Robotics and Automation, pages 6892–6903. IEEE, 2024.

[40] Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 17359–17369, 2025.

[41] Shivansh Patel, Xinchen Yin, Wenlong Huang, Shubham Garg, Hooshang Nayyeri, Li Fei-Fei, Svetlana Lazebnik, and Yunzhu Li. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards. arXiv:2502.08643, 2025.

[42] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. arXiv:2307.04577, 2023.

[43] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. arXiv:1709.10087, 2017.

[44] Rhoban. Placo: Rhoban planning and control. https://github.com/Rhoban/placo, 2025.

[45] Zhaole Sun, Kai Yuan, Wenbin Hu, Chuanyu Yang, and Zhibin Li. Learning pregrasp manipulation of objects from ungraspable poses. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 9917–9923. IEEE, 2020.

[46] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In IEEE International Conference on Robotics and Automation, pages 8112–8119, 2023.

[47] Wil Thomason, Zachary Kingston, and Lydia E Kavraki. Motions in microseconds via vectorized sampling-based planning. In IEEE International Conference on Robotics and Automation, pages 8749–8756, 2024.

[48] Eleftherios Triantafyllidis, Fernando Acero, Zhaocheng Liu, and Zhibin Li. Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network ROMAN. Nature Machine Intelligence, 5(9): 991–1005, 2023.

[49] Shuaijun Wang, Wenbin Hu, Lining Sun, Xin Wang, and Zhibin Li. Learning adaptive grasping from human demonstrations. IEEE/ASME Transactions on Mechatronics, 27(5):3865–3873, 2022.

[50] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. arXiv:2502.05855, 2025.

[51] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. arXiv:2503.02881, 2025.

[52] Jianglong Ye, Keyi Wang, Chengjing Yuan, Ruihan Yang, Yiquan Li, Jiyue Zhu, Yuzhe Qin, Xueyan Zou, and Xiaolong Wang. Dex1b: Learning with 1b demonstrations for dexterous manipulation. arXiv:2506.17198, 2025.

[53] Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Vtla: Vision-tactile-language-action model with preference learning for insertion manipulation. arXiv:2505.09577, 2025.

[54] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. arXiv:2304.13705, 2023.

[55] Zhigen Zhao, Liuchuan Yu, Ke Jing, and Ning Yang. Xrobotoolkit: A cross-platform framework for robot teleoperation. arXiv:2508.00097, 2025.

[56] Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Yitao Liang, Yaodong Yang, and Yuanpei Chen. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. arXiv:2502.20900, 2025.

[57] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan 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.